Response to reviewer JRbP

We appreciate the referee's recognition of the value of our paper, and insightful comments helpful to improve our work. In the following, we address the proposed weakness and questions.

My main suggestions to improve the overall quality and clarity of the work are to:

1. Provide all information in the README so that people with access to CRSP and Polygon.io can run it
2. Specify the versions of the dependencies, or better, use a dependency management system (e.g., poetry)
3. Use a .gitgnore file to remove any unnecessary files such as .DS_Store and other files/directories
4. The notebook folder contains .pdf and .py files instead of actual notebooks
5. Improve the overall organization of the repository

We thank the reviewer for the positive evaluation of our problem formalization, proposed method, and experimental results. We also appreciate the thoughtful feedback on reproducibility and the code repository.

Regarding the performance metrics, we understand that unusually strong results in financial settings warrant close scrutiny, especially given the potential for subtle implementation issues such as time leakage. We have carefully verified that all calculations in our pipeline, including return alignment, feature construction, and portfolio simulation, are free from look-ahead bias. The results are consistent across multiple training windows and market regimes, supporting the robustness of our findings.

We would also like to note that, while the performance in rank space appears impressive, the strategy faces real-world limitations. In particular, realizing the returns in rank space currently requires frequent intra-day rebalancing, which incurs significant transaction costs. As shown in Appendix G.3 and Figure 13(b), profitability declines significantly when transaction costs rise above 5 basis points. This limitation highlights a promising avenue for future research: developing more efficient execution strategies for rank-space returns using tools such as stochastic control or reinforcement learning.

With respect to reproducibility, we emphasize that in addition to open-sourcing the code, we have included detailed pseudocode in the appendix that is included to meet the NeurIPS's guideline "*if the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm*." The pseudocode outlines all key components of our methodology, including market decomposition, trading signal generation, portfolio construction, and backtesting.

We also note that working with proprietary datasets such as CRSP and Polygon.io introduces additional complexities: These data providers offer different versions of their datasets, and deriving accurate intra-day capitalization trajectories from these two data sets requires careful inspection, handling of inconsistent data (mostly arises from incorrect adjustment to stock split or dividend payment), and stock universe alignment, which cannot be entirely automated and require expert oversight.

Furthermore, the paper presents a broad set of experimental results and empirical observations, many of which involve different settings of model and hyper-parameters. As such, it is challenging to streamline all these experiments into a single auto-runnable script without substantial engineering overhead.

Despite the challenges above, we are committed to making our work as accessible and reproducible as possible. In response to the reviewer's suggestions, we have included a detailed README specifying (i) version requirements, (ii) expected data formats, (iii) directory structure, (iv) commands to run the code. Furthermore, we have added a Jupyter notebook that enables reproducing the portfolio performance in Figure 5 per reviewer's request. We will be continuing working on more Jupyter notebooks that enables reproducing other results in the paper.

We hope these clarifications and updates address the reviewer's concerns.

The paper is mostly clear and well-written, although it contains some typos and other issues that I report here.

Line 86: typo: "that characterize" -> "that characterizes"

Lines 91, 102, 142: missing ref to appendix, and the corresponding proofs seem to be missing

Line 88: "where c_{i,t} is the capitalization of stock i at day t" -> There is no c_{i,t}

Line 90: "R_{i,t} gives the capitalization rank of stock i." -> There is no R_{i,t}

We thank the review for careful examination and have made corresponding changes for the first two items in our revised manuscript.

Regarding the review's comments on lines 88 and 90: $c_{i,t}$ refers to the capitalization of stock i at time t and is the generic form of $c_{I_{(k),t},t}$ and $c_{I_{(k),t-1},t-1}$ in the equality of equation (3.5). Similarly, $R_{i,t}$ denotes the rank in capitalization of stock i at time t and serves as the inverse map of $I_{(k),t}$. This notation is used throughout the manuscript, particularly in the formulation of our intra-day rebalancing strategy (e.g., Equation 3.13).

The rank space is partially original but not completely new. Indeed, there is extensive literature on asset pricing that utilizes cross-sectional sorts based on market capitalization, dating back to at least 1992 (https://onlinelibrary.wiley.com/doi/10.1111/j.1540-6261.1992.tb04398.x). The related works may be improved by mentioning that cross-sectional sorts are standard in the asset pricing literature, and here, this knowledge is embedded into the data representation for deep networks.

We respectfully disagree with the reviewer's judgement on the originality of our work by comparing our work to cross-sectional sorts in asset pricing literature. Our contribution differs in both scope and motivation from traditional cross-sectional sorting. While such sorting is typically used for asset pricing tests or factor portfolio construction, our approach embeds this ranking as a structural transformation of the data space, grounded in a theoretical framework that yields new insights into market dynamics and statistical arbitrage performance.

Specifically, our rank-space formulation has deep theoretical motivations that supports two critical empirical evidence: (i) a more structured covariance spectrum with enhanced signal-to-noise properties (Figure 3), and (ii) a formal derivation of robust mean-reversion dynamics in residual returns (Figure 4). Below, we briefly summarize these theoretical foundations.

First, in our ongoing theoretical paper (ref [16]), we rigorously show why transforming market data into rank space systematically clears the market structure and enhances the signal-to-noise ratio. Briefly, consider an indication matrix $\{\theta_t\}_{1\le i\le N, 1\le j\le K} = \chi(c_{i,t} = c_{(k),t})$[3] that maps the stock from name space to rank space based on their ranks in capitalization. Define the occupation rate $\Theta = \frac{1}{T}\sum_{t=1}^{T}\theta_t$ which transforms the covariance matrix of stock returns from name space (Σ) into rank space ($\Theta^{\mathrm{T}}\Sigma\Theta$). Under standard ergodic market dynamics assumptions, we theoretically demonstrate that this mapping yields a more structured eigenvalue spectrum, characterized by an enhanced gap between the largest eigenvalue and the smaller eigenvalues. Such an eigenvalue structure inherently enhances the market structure and facilitates the market decomposition, directly supporting the empirical findings presented in Figure 3 of our manuscript.

Second, the pronounced mean-reversion of residual returns in rank space (Figure 4) also has deep theoretical anchors. In the appendix B of the revised manuscript, we justify such behavior under the framework of a hybrid-Atlas model. Starting from the market dynamics in name space where the residual returns in name space are modeled by Brownian motions (equation B.1), theorem B.3 shows that transforming into rank space introduces additional local time terms for the residual returns in rank space, significantly promoting the mean-reversion in the residual returns in rank space.

Therefore, these results go beyond empirical sorting and demonstrate that rank space is not merely a preprocessing step, but rather a theoretically motivated representation of market structure. We hope this clarification better highlights the novelty of our approach.

---

[3] $c_{i,t}$ and $c_{(k),t}$ are defined in equation (3.5) in the main text. $\chi(\cdot)$ is the indication function that takes 1 if $(\cdot)$ holds otherwise 0.

Line 87: This definition reads like a contradiction: "daily return on rank k at day t in the continuous-time limit." Do you mean daily returns for the rank k that are computed by continuously compounding the returns of the stocks that enter and exit rank k during the day?

We thank the reviewer for bringing up this clarification request. The "daily return on rank k at day t in continuous-time limit is defined in equation (3.5). Specifically, Equation (3.5) defines this return can be viewed as the discrete analog of the derivative of log-capitalization process in rank space $d \log Z_k(t)$ defined in appendix B in our revised manuscript, accounting for the fact that the identity of the stock occupying rank k may change over time.

The name space is affected by several outliers in returns, while in rank space the returns should not exhibit many outliers. For instance, if there is a jump in the price of a stock in rank k, then it would move to a higher capitalization rank k', and the return of rank k would not increase too much. This difference may be significant in understanding why rank space works so much better than name space. How have returns been standardized to train the networks in name space?

We thank the reviewer for bringing up this very intriguing question. Returns in both name space and rank space have been standardized to before being used to train the neural networks.

In fact, the return differs between name space and rank space when there is rank switching in capitalization, which is the key for the success of our proposed strategy here. The difference leads to robust market structure and enhanced mean-reversion of residual returns, as supported by both empirical evidence (Figure 3 and 4) and theoretical justifications in our responses above.

The comparison between name space and rank space may be unfair, in the sense that name space only uses cumulative returns. Instead, rank space implicitly uses both name returns and market capitalization. A better comparison would be to compare the current model trained on rank space with a model trained in name space, using cumulative returns as well as market capitalization as additional features. Do I understand correctly that the current model in name space never sees the market capitalization during training?

We thank the reviewer for this thoughtful and important question. To clarify: yes, in the reported results, the model trained in name space only uses cumulative returns as input features and does not observe market capitalization during training.

We conducted an additional experiment where we augmented the input to the name-space model with the full time series of market capitalizations. Interestingly, this modification led to a deterioration in portfolio performance compared to the original name-space model. This result reinforces a central message of our work: the effectiveness of deep learning in financial modeling is not solely determined by access to more raw information, but rather by how the data is encoded to reflect meaningful market structure. Representing the data in rank space allows the model to exploit the underlying dynamics more effectively, which is crucial for robust generalization and performance.

I'm missing a sense of the generalizability of the main conclusion that using rank space may improve learning in noisy financial environments. Benchmarking tests of rank versus name space with other architectures are missing. How much of the improvement between name and rank space is due to the specific architecture used here?

We thank the reviewer for raising this important question about generalizability and the role of model architecture. To assess whether the performance gain observed in rank space is dependent on the specific architecture used, we conducted a comparative analysis using both a parametric model (described in Appendix E) and neural networks, applied in both name space and rank space and the results are shown in Figure 5:

(i) Panel (a) vs. Panel (b): These compare the portfolio performance of a parametric model trained in name space and rank space, respectively. The improved performance in panel (b) demonstrates that even a simple model benefits from transforming market data into rank space.

(ii) Panel (d) vs. Panel (e): These compare neural network-based models trained in name space and rank space. Once again, the rank-space model significantly outperforms its name-space counterpart.

These cross-architecture comparisons indicate that the improvements associated with rank-space representation are not confined to the neural network architecture. Rather, the benefits appear to stem from the representation itself, which facilitates learning by reducing noise and highlighting more structured market dynamics. This suggests that the advantages of using rank space are robust and generalizable across model classes.

The paper title should be in title case, not sentence case -> Deep Statistical Arbitrage: Name Space Versus Rank Space

Table 1 in Appendix E goes out of page

We appreciate the reviewer's careful examination to help us improve the format of our manuscript. We have made corresponding corrections in the revised manuscript.

Deep Statistical Arbitrage: Name Space versus Rank Space

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Equity market dynamics are conventionally investigated in name space, where stocks are indexed by company names. However, this perspective often suffers from high volatility and a low signal-to-noise ratio, which poses challenges for effective learning by deep neural networks (DNNs). In contrast, by indexing stocks by their ranks in capitalization, we gain a distinct and more structured view of market behavior in rank space. In this work, we demonstrate that DNNs achieve superior performance in statistical arbitrage when operating in rank space compared to name space. This performance gain is driven by more robust market representations and enhanced mean-reverting properties of residual returns in rank space, which facilitate more efficient learning. Our findings highlight the critical role of domain-informed data transformation in improving deep learning performance in noisy financial environments.

1 Introduction

In equity markets, stocks are conventionally labeled by fixed company identifiers (company names), a perspective that often suffers from high volatility and a low signal-to-noise ratio, hindering effective learning by deep neural networks (DNNs).

As an alternative, we consider a rank-space representation, where stocks are indexed by their ranks in capitalization rather than fixed company identifiers. In this view, we focus on the stock at a given rank in capitalization while the corresponding company name may change. This formulation offers a more structured and stable view of market dynamics. We refer to a market labeled by fixed company identifiers (company names) as the *market in name space*, and one labeled by ranks in capitalization as the *market in rank space*.

A structured market dynamics is crucial for the performance of statistical arbitrage, a trading strategy that exploits temporary under- and over-pricing of stocks. Specifically, statistical arbitrage constructs market-neutral portfolios that reproduce the residual returns – returns not explained by market factors – and generates profits if these residual returns are mean-reverting.

In this paper, we show that operating statistical arbitrage in rank space by DNNs significantly outperforms the name-space counterpart. Our DNN-based rank-space portfolios achieve an average annual return 35.68% and an average Sharpe ratio of 3.28 from 2007 to 2022, accounting for a 2-basis-point transaction cost. In contrast, applying the same DNNs in name space yields negligible returns over the same period.

Further comparison shows that DNN-based rank-space portfolios not only exploit the importance of mean-reversion during mean-variance optimization, but also implement more intelligent strategies than traditional parametric model by applying flexible leverage and minimizing carry risk. We attribute these improvements to two key advantages of rank space: (i) a more robust and stable

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

representation of market structure, and (ii) enhanced mean-reverting behavior in residual returns. Together, these properties enable DNNs to extract actionable signals from noisy financial data.

Our findings highlight the importance of domain-informed data representations in financial machine learning. In particular, they show how appropriate input transformation can significantly improve the learning efficiency and performance of deep models in complex, noisy environments like equity markets[9, 23].

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 formulates the framework for statistical arbitrage in both name space and rank space. Section 4 presents our empirical results using U.S. equity date. Section 5 concludes with a discussion of implications and future directions.

2 Related work

Our results add to the burgeoning literature on machine learning applications in finance. Gu *et al.* systematically compare various machine learning methods for predicting stock returns[13]. Acero *et al.* utilize deep reinforcement learning for portfolio managements[1]. Araci explores the correlation between financial news and stock returns by BERT language model[2]. Horvath *et al.* apply deep neural networks to calibrate volatility surface in fixed-income markets[17].

Our research advances the understanding of statistical arbitrage. While this problem can be framed as a stochastic control problem solvable via the Hamilton-Jacobi-Bellman (HJB) equation[10, 20, 25, 29], practical applications are often limited by calibration challenges and the absence of selection mechanism in diverse markets. To address these issues, Avellaneda and Lee[3] propose a pragmatic trading strategy based on Ornstein–Uhlenbeck (OU) processes, later refined by Yeo and Papanicolaou[28]. More recently, Mulvey *et al.*[22] combine the HJB framework with feed-forward networks, while Kim *et al.*[19] apply deep reinforcement learning to optimize pair trading strategies. Guijarro-Ordonez *et al.*[14] provide a comprehensive survey of deep learning models for statistical arbitrage.

Our paper also enriches the emerging literature on rank-based market models. Fernholz *et al.*, in their seminal work on stochastic portfolio theory, introduce the concept of functionally generated portfolios in both name and rank spaces[11]. Building on this foundation, Banner *et al.*[7] and Ichiba *et al.*[18] study the dynamics of rank capitalizations and gap processes in Atlas and hybrid Atlas models, respectively. Healy *et al.*[16] systematically investigate the market structure in rank space by Θ-transformation.

3 Formulation

This section formulates statistical arbitrage in name and rank space. A schematic overview is shown in Fig. 1, detailed algorithmic pseudocode is provided in Appendix A.3, and the code is available at `https://github.com/Infi-Yingfei-Li/stats-arb-rank-space`.

3.1 Market decomposition

3.1.1 Name space

In a market consisting of N stocks, we denote the dividend-adjusted return[1] on stock i at trading day t by $r_{i,t}$. We adopt a standard factor model for stock returns,

$$r_t - r_f = \beta_t F_t + \epsilon_t, \qquad t = 1, 2, ..., T. \tag{3.1}$$

Here, $r_t = \{r_{i,t}\}_{i=1}^N \in \mathbb{R}^N$ are the dividend-adjusted daily return, $r_f \in \mathbb{R}$ is the risk-free rate[2], $F_t \in \mathbb{R}^{K\times 1}$ are the underlying factors, $\beta_t \in \mathbb{R}^{N\times K}$ are the corresponding loadings on K factors, and $\epsilon_t \in \mathbb{R}^N$ are the residual returns.

Without loss of generality, these factors can be written as portfolios of stocks,

$$F_t = \omega_t(r_t - r_f), \tag{3.2}$$

[1] The daily return of a stock that accounts for both price changes and dividend payments.

[2] The investment return with zero-risk financial loss, chosen as the rate of 1-month U.S. treasury bill here.

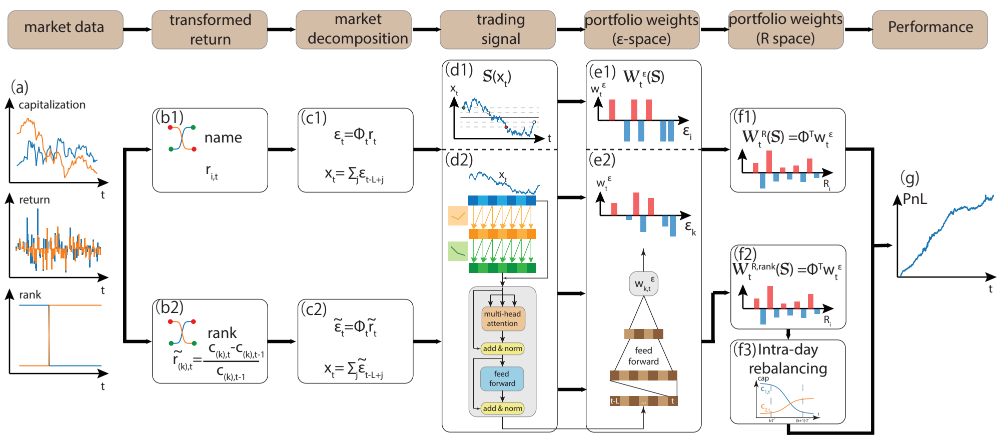

Figure 1: Schematic of the statistical arbitrage algorithm in name space and rank space.

where $\omega_t \in \mathbb{R}^{K \times N}$ specifies the factor portfolio weights. Combining (3.1) and (3.2) yields

$$r_t - r_f = \beta_t \omega_t (r_t - r_f) + \epsilon_t \Rightarrow \epsilon_t = (I - \beta_t \omega_t)(r_t - r_f) := \Phi_t (r_t - r_f) \tag{3.3}$$

where $\Phi_t := (I - \beta_t \omega_t)$ defines a linear transformation from returns to residual returns. Each $\epsilon_{i,t}$ can thus be interpreted as the return of a tradable portfolio with weights given by the i-th row of Φ_t.

Consequently, we refer to the space spanned by r_t as the *name equity space*, and the space spanned by ϵ_t as the *name residual space*. We denote the portfolio weights in name equity space as $w_t^{R,\text{name}}$ and portfolio weights in name residual space as $w_t^{\epsilon,\text{name}}$. These are related by

$$w_t^{R,\text{name}} = \Phi_t^T w_t^{\epsilon,\text{name}} \tag{3.4}$$

Importantly, given any $w_t^{\epsilon,\text{name}}$, the derived $w_t^{R,\text{name}}$ are market neural (proofs in the Appendix C).

3.1.2 Rank space

We begin by introducing the key variables that characterize the market dynamics in rank space: the daily return on rank k at day t in the continuous-time limit, defined as

$$\tilde{r}_{(k),t} := \frac{c_{(k),t} - c_{(k),t-1}}{c_{(k),t-1}} = \frac{c_{\mathcal{I}_{(k),t},t} - c_{\mathcal{I}_{(k),t-1},t-1}}{c_{\mathcal{I}_{(k),t-1},t-1}}, \tag{3.5}$$

where $c_{i,t}$ is the capitalization of stock i at day t, $c_{(k),t}$ is the capitalization of the stock occupying the k-th rank in descending order at day t. $\mathcal{I}_{(k),t}$ represents the company occupying the rank k at day t, and conversely, $\mathcal{R}_{i,t}$ gives the capitalization rank of stock i. Our definition of return in rank space is motivated by the log capitalization process in the hybrid-Atlas model in Appendix B[7, 18].

Importantly, $\tilde{r}_t$ does not necessarily correspond to direct financial quantity, as $\mathcal{I}_{(k),t}$ and $\mathcal{I}_{(k),t-1}$ may differ – meaning the stock occupying rank k can change between days. To realize $\tilde{r}_t$ in practice, we develop an intra-day rebalancing strategy, detailed in Appendix D.

Following the construction in name space, we assume a factor model for $\tilde{r}_t$,

$$\tilde{r}_t - r_f = \tilde{\beta}_t \tilde{F}_t + \tilde{\epsilon}_t, \tag{3.6}$$

where $\tilde{r}_t = \{r_{(k),t}\}_{k=1}^N \in \mathbb{R}^N$, $\tilde{\beta}_t \in \mathbb{R}^{N \times K}$, $\tilde{F}_t \in \mathbb{R}^{K \times 1}$, and $\tilde{\epsilon}_t \in \mathbb{R}^N$. Analogously, we model the factors as portfolios of rank returns

$$\tilde{r}_t - r_f = \tilde{\beta}_t \tilde{\omega}_t (\tilde{r}_t - r_f) + \tilde{\epsilon}_t \Rightarrow \tilde{\epsilon}_t = (I - \tilde{\beta}_t \tilde{\omega}_t)(\tilde{r}_t - r_f) := \tilde{\Phi}_t (\tilde{r}_t - r_f), \tag{3.7}$$

where $\tilde{\Phi}_t := (I - \tilde{\beta}_t \tilde{\omega}_t)$.

We refer to the space spanned by $\tilde{r}_t$ as *rank equity space* and the space spanned by $\tilde{\epsilon}_t$ as *rank residual space*, mirroring our definition in the name space. Let $w_t^{R,\text{rank}}$ and $w_t^{\epsilon,\text{rank}}$ denote the portfolio weights in rank equity space and rank residual space, respectively, related by

$$w_t^{R,\text{rank}} = \Phi_t^T w_t^{\epsilon,\text{rank}}, \tag{3.8}$$

As in name space, any $w_t^{\epsilon,\text{rank}}$ generates a market-neutral portfolio $w_t^{R,\text{rank}}$ (proof in Appendix C).

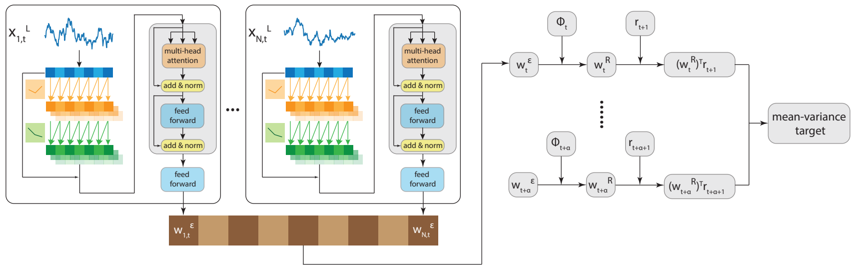

Figure 2: Schematic for the architecture of deep neural networks in both name and rank space.

3.2 Trading signals and portfolio weights

After computing the residual returns ϵ_t[3], we derive the cumulative residual returns over a look-back window of length L:

$$x_t^L = (x_{t-L+1}, x_{t-L+2}, ..., x_t), \tag{3.9}$$

where $x_{t-L+\alpha} = \sum_{j=1}^{\alpha} \epsilon_{t-L+j}, \quad \alpha = 1, 2, \ldots, L.$

We adopt DNNs $\mathcal{N} : x_t^L \to w_t^{\epsilon|\text{NN, name/rank}}$ as a data-driven method to calculate portfolio weights in residual space w_t^ϵ for both name space and rank spaces. The DNNs consist of convolutional layers to capture local patterns, followed by transformer encoder layers to model global dependencies. The neural networks are trained via mean-variance optimization,

$$\begin{aligned} \text{Maximize}_{\mathcal{N}(\cdot)} \quad & \mathbb{E}[(w_t^{R|\text{NN,name/rank}})^T (r_{t+1} - r_f)] - \gamma \text{Var}[(w_t^{R|\text{NN,name}})^T (r_{t+1} - r_f)] \\ \text{s.t.} \quad & w_t^{R|\text{NN,name/rank}} = \frac{\Phi_t^T w_t^{\epsilon|\text{NN,name/rank}}}{||\Phi_t^T w_t^{\epsilon|\text{NN,name/rank}}||_1} \\ & w_t^{\epsilon|\text{NN,name/rank}} = \mathcal{N}(x_t^L), \end{aligned} \tag{3.10}$$

where γ is the risk-aversion factor. We show a schematic for our DNNs architecture in Fig. 2 with architecture and implementation details in Appendix A.2.

For comparison, we benchmark the DNN performance against a classical parametric model based on Ornstein-Uhlenbeck (OU) process[3, 28], with the model formulation and execution details in Appendix E.

3.3 Intraday rebalancing

The portfolio weights calculated in the rank space are assigned to artificial financial instruments that yield rank returns in continuous-time limits defined in (3.5). To make the constructed portfolio practically implementable, it is necessary to convert these portfolio weights into stock-based portfolios in name space. To address the issue, we propose an intraday rebalancing mechanism.

Formally, given the predetermined portfolio weights in rank equity space $\{w_{(k),t}^{\text{rank}}\}_{k=1}^N$ before the market opening, our goal is to rebalance the portfolio at fixed time intervals of $\mathcal{T}$ minutes such that the portfolio becomes $\{w_{(k),t}^{\text{rank}}(1 + \tilde{r}_{(k),t+1})\}_{k=1}^N$ by market close, at the sacrifice of additional costs. To facilitate this discussion, we introduce two processes:

(i) $w_{(k),t+\tau}^{\text{rank}}$: the dollar-valued portfolio weight for rank k at time $t + \tau$;

(ii) $w_{i,t+\tau}^{\text{name}}$: the dollar-valued portfolio weight for stock i at time $t + \tau$.

Here, t denotes the daily time tick and τ represents the intraday time tick. For instance, for $t =$ Jan, 3rd, 2022 (end of the day) and $\tau = 45$ minutes, $t + \tau$ refers to Jan, 4th, 2022 00:45 AM, and $t + 1$ refers to the end of the day on Jan, 4th, 2022.

[3] For simplicity, we take a unified notation ϵ_t to denote the residual returns in both name and rank space in the following discussions unless otherwise specified.

$w^{\text{rank}}_{(k),t+\tau}$ is the portfolio weight on the k-th rank that evolves strictly based on the rank returns in continuous-time limit,

$$w^{\text{rank}}_{(k),t+\tau} = w^{\text{rank}}_{(k),t}(1+\tilde{r}_{(k),t+\tau}), \tag{3.11}$$

where $\tilde{r}_{(k),t+\tau} = \frac{c_{(k),t+\tau}}{c_{(k),t}} - 1$. Here, $c_{(k),t+\tau}$ denotes the capitalization at k-th rank at time $t+\tau$.

In contrast, $w^{\text{name}}_{i,t+\tau}$ is the portfolio weight on the i-th stock, evolving according to the following rules:

(i) Between the rebalancing interval when $t + j\mathcal{T} < t + \tau \leq t + (j+1)\mathcal{T}, j \in \mathbb{N}$,

$$w^{\text{name}}_{i,t+\tau} = w^{\text{name}}_{i,t+(j\mathcal{T})^+} \times \frac{c_{i,t+\tau}}{c_{i,t+(j\mathcal{T})^+}}, \tag{3.12}$$

where $(j\mathcal{T})^+ := \lim_{\delta\downarrow 0}(j\mathcal{T}+\delta)$.

(ii) At the re-balancing points when $\tau = ((j+1)\mathcal{T})^+, j \in \mathbb{N}$, adjust the portfolio weights via active trading such that

$$w^{\text{name}}_{i,t+((j+1)\mathcal{T})^+} = \sum_{k=1}^{n} w^{\text{rank}}_{(k),t+(j+1)\mathcal{T}} \mathbf{1}_{\{\mathcal{R}_{i,t+(j+1)\mathcal{T}}=k\}}. \tag{3.13}$$

In other words, we carry out the conversion of portfolio weights between name space and rank space at the rebalancing points $\tau = ((j+1)\mathcal{T})^+, j \in \mathbb{N}$ through active trading. Notably, the value on the trading book before trading at $t+(j+1)\mathcal{T}$ is $\sum_{k=1}^{N} w^{\text{name}}_{i,t+(j+1)\mathcal{T}}$, while the desired value immediately after trading is $\sum_{i=1}^{N} w^{\text{name}}_{i,t+(j+1)\mathcal{T}^+} = \sum_{k=1}^{N} w^{\text{rank}}_{(k),t+(j+1)\mathcal{T}}$. The two values are not necessarily equal when there exists switching of ranks in capitalization between $t + j\mathcal{T}$ and $t + (j+1)\mathcal{T}$ (elaborated by a case study in Appendix D and by the hybrid-Atlas model in Appendix B). Consequently, the cost of the active trading at $t + (j+1)\mathcal{T}$ involves two components,

$$\begin{aligned} \text{cost}(t+(j+1)\mathcal{T}^+; w^{\text{rank}}_{(k),t}) =& (\sum_{i=1}^{N} w^{\text{name}}_{i,t+(j+1)\mathcal{T}^+} - \sum_{k=1}^{N} w^{\text{rank}}_{(k),t+(j+1)\mathcal{T}}) \\ &+ \eta \sum_{i=1}^{N} |w^{\text{name}}_{i,t+(j+1)\mathcal{T}^+} - w^{\text{name}}_{i,t+(j+1)\mathcal{T}}|, \end{aligned} \tag{3.14}$$

where η is the transaction cost factor. Here, the cost at time $t + (j+1)\mathcal{T}^+$ depends on the portfolio weights assigned at the beginning of the trading day, $w^{\text{rank}}_{(k),t}$, because the intraday portfolio weights $w^{\text{name}}_{i,t+\tau}$ and $w^{\text{rank}}_{(k),t+\tau}$ are recursively governed by the system dynamics ((3.11), (3.12), and (3.13)) from the initial condition $w^{\text{rank}}_{(k),t}$. We highlight this dependence by including the $w^{\text{rank}}_{(k),t}$ as a parameter for the cost in (3.14). We refer to the first term in (3.14) as latency cost, the second term as cost from the bid-ask spread, with their sum representing the total transaction cost. The terminology are rationalized in the case study in Appendix D.

The precise implementation of the intraday rebalancing is summarized in Appendix D Algorithm 4 along with a schematic in panel (f3) in Fig. 1. For our backtesting, we primarily use $\eta = 2$ basis points to account for the cost from the bid-ask spread. This setting approximately corresponds to a 5-10 cents bid-ask spread for our investment universe, the top 500 stocks in the U.S. equity market.

3.4 Backtesting

We evaluate the portfolio performance by calculating the historical profit-and-loss (PnL) V_t and the Sharpe ratio.

For portfolios in name space,

$$V_{t+1} = (1 + r_{f,t+1}) \times (V_t - \sum_i \Lambda w_{i,t} V_t - \text{TC}) + \sum_i \Lambda V_t w_{i,t}(1 + r_{i,t+1}), \tag{3.15}$$

where $\Lambda = 1$ is the leverage, $r_{f,t+1}$ is the risk-free rate during the trading day $t+1$, and w_t are normalized by l_1 norm. The transaction cost is given by $\text{TC} = \eta \sum_i \Lambda |V_t w_{i,t} - V_{t-1} w_{i,t-1}(1+r_{i,t})|$, where η is the transaction cost factor, set to 2 basis points.

For portfolios in rank space, the PnL evolves as

$$\begin{aligned} V_{t+1} =&(V_t - \sum_{k=1}^{N} w_{(k),t}^{R,\text{rank}})(1+r_{f,t+1}) + \sum_{k=1}^{N} \Lambda V_t w_{(k),t}^{R,\text{rank}}(1+\tilde{r}_{(k),t+1}) \\ &- \sum_{j:t<t+j\mathcal{T}\leq t+1} \text{cost}(t+j\mathcal{T}^{+}; \Lambda V_t w_{(k),t}^{R,\text{rank}}), \end{aligned} \tag{3.16}$$

where the last term accounts for the transaction costs due to intraday rebalancing defined in (3.14), where we substitute generic portfolio weights $w_{(k),t}^{\text{rank}}$ in (3.14) into specific portfolio weights $\Lambda V_t w_{(k),t}^{\text{R, rank}}$ in (3.16).

4 Empirical results for the U.S. equities

4.1 Data and experimental setup

We collect dividend-adjusted daily return, price, shares outstanding, and capitalizations for the U.S. securities from Center for Research in Security Prices (CRSP), covering January 1990 to December 2022. Intraday price data at 1-minute resolution from January 2005 to December 2022 are obtained from *Polygon.io*. We construct intraday capitalization data by combining CRSP shares outstanding with *Polygon.io* intraday prices. The one-month Treasury bill rate from the Kenneth French Data Library is used as the risk-free rate r_f. Further implementation details are provided in Appendix A.

4.2 Market structure: name space versus rank space

We begin by comparing the market dynamics in name space and rank space, highlighting two key advantages of rank space: (i) a more structured market dynamics in rank space, and (ii) a more enhanced mean-reverting behavior of residual returns in rank space – both critical motivations for operating statistical arbitrage in rank space.

First, we show the market capitalization across ranks, averaged over five-year window from 1991 to 2022 in Fig. 3(a), revealing a stable distribution in rank space[11]. A principal component analysis (PCA) on the correlation matrix suggests significantly larger leading eigenvalue in rank space compared to that in name space (Fig. 3(b)), implying that a greater proportion of market variance is captured by the first eigenvector in rank space. This points to a more structured and concentrated market dynamic in rank space.

More importantly, the single-factor structure in rank space substantially simplifies market decomposition. We present the empirical eigenvalue spectra of the correlation matrix in both spaces across different periods in Fig. 3(c1-c6) and (d1-d6). In name space, several eigenvalues exceed the Marchenko–Pastur upper bound [4], indicating a multi-factor market and consequently low signal-to-noise ratio(Fig. 3(c1-c6)). In contrast, rank space exhibits a sharp bulk-edge separation with a dominant single factor (Fig. 3(d1-d6)). This clearer structure enables more well-defined separation of market factors from residuals during market decomposition.

Second, we observe markedly faster mean-reversion of residual returns in rank space, critical for statistical arbitrage. We quantify mean-reversion by fitting cumulative residual returns x_t^L to an OU process and extracting the mean-reversion time τ. Fig. 4 shows the empirical distribution of τ over five-year windows from 1991 to 2022. In name space (Fig. 4(a1–a6)), the distribution is heavy-tailed toward large τ, reflecting slower mean reversion. In contrast, rank space (Fig. 4(b1–b6)) exhibits a sharper concentration in the fast mean-reverting regime, with fewer instances of slow mean-reversion ($\tau > 30$ days, shaded area). A non-parametric analysis in Appendix F yields consistent results.

4.3 Portfolio performance

We present the PnL V_t in Fig. 5 with portfolio weights w_t^R are calculated by four scenarios: (i) the parametric benchmark model in name space in panel (a), (ii) the parametric benchmark model in rank space in panels (b,c), (iii) DNNs in name space in panel (d), and (iv) DNNs in rank space in panels (e, f). The corresponding Sharpe ratios from 2016 to 2022 are summarized in Fig. 5(g, h), with year-by-year statistics provided in Appendix G.1 (without transaction costs: Table 1; with transaction costs: Table 2).

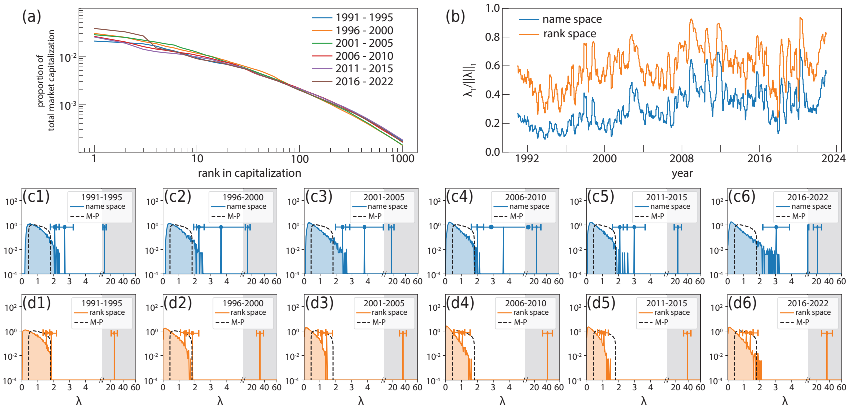

Figure 3: **Market structure in name space versus rank space.** **(a)** Proportion of total market capitalization versus ranks in capitalization. **(b)** The principal eigenvalue of the correlation matrices of r_t in name space (blue) and for $\tilde{r}_t$ rank space (orange). **(c, d)** The empirical probability distribution density of the eigenvalue spectrum of the correlation matrices versus Marchenko-Pastur distribution. The market exhibits a more structured correlation in rank space compared to name space.

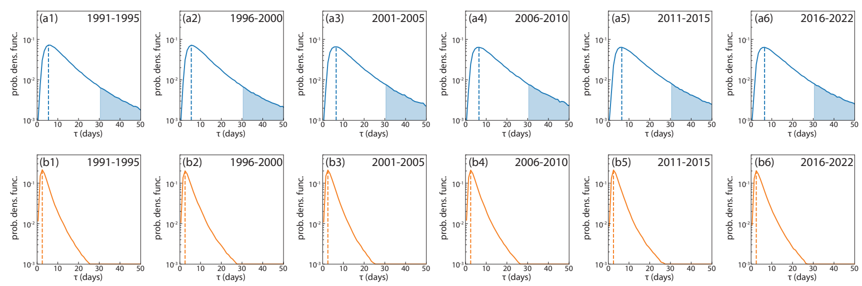

Figure 4: **Mean-reverting time τ in name space versus rank space.** The τ is evaluated by fitting the cumulative residual return x_t^L to an Ornstein–Uhlenbeck(OU) process. **(a1-a6)** The empirical distributions of mean-reverting time τ in name space, with maximum empirical probability at ~ 6 days (vertical dashed lines). **(b1-b6)** The empirical distributions of τ in rank space, with maximum empirical probability at ~ 2.5 days (vertical dashed lines). The residual returns in rank space show faster mean-reverting behavior, favorable for statistical arbitrage.

The traditional statistical arbitrage strategy in name space using the parametric model exhibits diminishing profitability after the 2010s. In rank space, the parametric model yields mixed results: while initial performance without transaction costs appears attractive (Fig. 5(b)), accounting for transaction costs ($\eta = 0.0002$) leads to a monotonic decline in PnL (Fig. 5(c)). This stark contrast highlights the substantial cost of realizing continuous-time rank returns $\tilde{r}_t$ via intraday rebalancing.

To address these challenges, we leverage DNNs to better exploit market patterns, particularly in rank space. The performance of DNNs in name and rank spaces diverges sharply. In name space, DNNs fail to improve returns or Sharpe ratios (Fig. 5(a, d); Appendix G.1, Table 1, Table 2). In contrast, in rank space, DNNs substantially enhance portfolio performance, achieving an average annual return of 35.68% and an average Sharpe ratio of 3.28 from 2007 to 2022 (Fig. 5(h); Table 2), even after accounting for transaction costs. This success is driven by the effective exploitation of mean-reversion behavior in rank space by DNNs that yields an average annual return of 206.49% and an average annual Sharpe ratio of 9.04 without transaction costs (Appendix G.1, Fig. 5(g), Table 1), sufficient to offset the substantial costs in intraday rebalancing to realize $\tilde{r}_t$.

Further characterization of the portfolios is provided in Appendix G. Appendix G.1 presents annualized performance statistics, Appendix G.2 examines market and dollar neutrality, Appendix G.3 discusses dependence on transaction costs, and Appendix G.4 analyzes the role of rank-swapping timescales on portfolio performance.

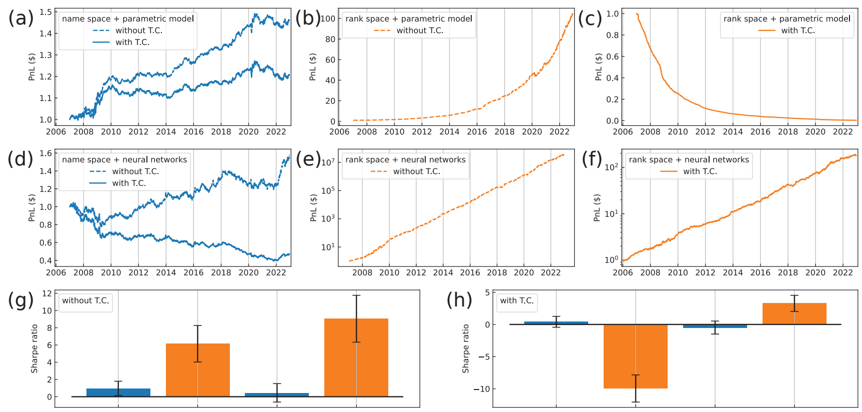

Figure 5: **Summary of portfolio performance.** The PnL dynamics V_t are computed using (3.15) in name space and (3.16) in rank space. **(a)** PnL using portfolio weights from the parametric model in name space; dashed and solid lines represent results without and with transaction costs, respectively. **(b, c)** PnL using portfolio weights from the parametric model in rank space; dashed/solid lines represent results without/with transaction costs in panel (b)/(c). **(d)** Same as (a), but using portfolio weights derived from DNNs in name space. **(e, f)** Same as (b, c), but using portfolio weights from DNNs in rank space. **(g, h)** Average Sharpe ratio without/with transaction costs shown in panel (g)/(h). Notably, portfolios derived from DNNs in rank space perform significantly better than those in name space.

4.4 The intelligence inside the neural networks

To understand the outperformance of DNNs compared to the benchmark parametric model in rank space, we analyze the relationship between the input (the trajectories of the cumulative residual return x_t^L) and the output (the portfolio weights in residual space w_t^ϵ).

We parameterize x_t^L by two key variables: the deviation from long-term average $\frac{x_t-\mu}{\sigma}$, and the mean-reverting time τ, following the trading signal suggested by the parametric model((E.4)). Each x_t^L thus corresponds to a point in the plane spanned by $\frac{x_t-\mu}{\sigma}$ and τ, color-coded by w_t^ϵ (Fig. 6).

We evaluate four scenarios: (i) parametric model in name space (Fig. 6(a)), (ii) DNNs in name space (Fig. 6(b)), (iii) parametric model in rank space (Fig. 6(d)), and (iv) DNNs in rank space (Fig. 6(e)). We also report the average holding periods before liquidation for each method (Fig. 6(c, e)).

Compared to the parametric benchmark, the DNNs demonstrate more sophisticated trading behavior in rank space. Despite operating directly on raw cumulative return trajectories, the DNNs successfully uncover the significance of mean-reversion through mean-variance optimization (Fig. 6(e)). In addition, the DNNs improve the execution strategy along three dimensions:

(i) Variable leverage on deviations. The DNNs assign higher leverage to positions with larger normalized deviations $\frac{x_t-\mu}{\sigma}$, enhancing profit potential during significant market moves (Fig. 6(e)).

(ii) Flexible opportunity thresholds. Rather than relying on rigid mean-reversion time (τ) cutoffs as in the parametric model (Fig. 6(a, d)), DNNs embrace a broader range of trading opportunities while concentrating investments in trajectories with fast mean-reversion (τ small) (Fig. 6(b, e)).

(iii) Shorter holding periods. DNNs reduce average holding periods to around 5 days, compared to approximately 10 days under the parametric model (Fig. 6(c, e)). This minimizes carry-over risk, which is particularly important when employing variable leverage across positions.

We also emphasize the critical role of market data preprocessing. While the DNNs achieve substantial improvements in rank space, they fail to converge or deliver gains in name space (Fig. 7; Fig. 5(d)). Despite both input spaces being derived from similar capitalization data, strategic reorganization into rank space dramatically improves training efficiency (Fig. 7) and portfolio performance (Fig. 5). This illustrates the importance of domain-informed data transformations in financial machine learning:

appropriate restructuring of the input space can substantially enhance the learning efficiency and performance of deep models in complex, noisy environments like equity markets.

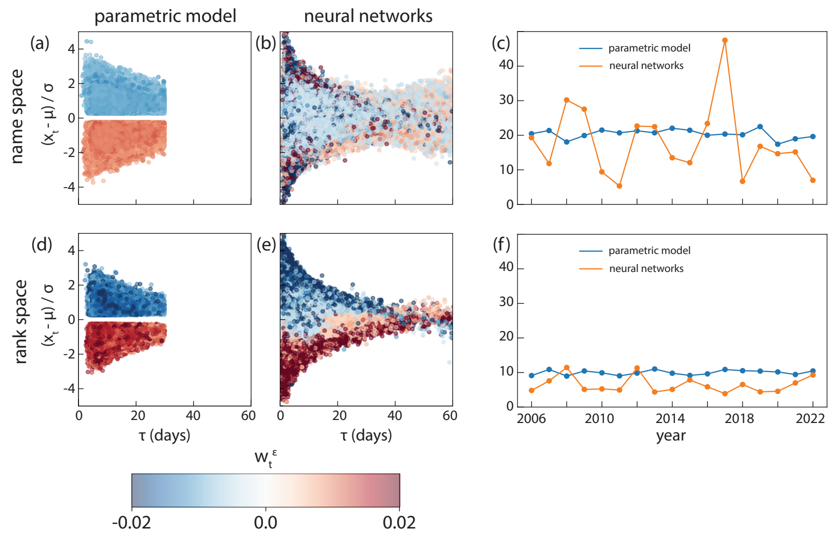

Figure 6: **Portfolio weights in residual space: parametric model versus neural networks. (a, b, d, e)** We illustrate the behavior of both the parametric model and neural networks by analyzing the relationship between the input, the trajectories of cumulative residual returns x_t^L, and the output, the portfolio weights in residual space, w_t^ϵ. The input x_t^L are parameterized by two variables: (i) deviation from long-term average $\frac{x_t-\mu}{\sigma}$, and (ii) mean-reverting time τ. Each x_t^L thus corresponds to a point in the plane spanned by $\frac{x_t-\mu}{\sigma}$ and τ, color-coded by w_t^ϵ. This analysis is performed with w_t^ϵ calculated by four scenarios: (a) the parametric model in name space; (b) neural networks in name space; (d) the parametric model in rank space; (e) neural networks in rank space. **(c, f)** Panel (c) and (f) show the average holding time for portfolios derived in name space and rank space, respectively. The DNN-derived portfolios exhibit more intelligence in terms of variable leverage, flexible opportunity thresholds, and shorter holding periods.

5 Limitations and discussion

Our current strategy is sensitive to transaction costs, particularly due to the frequent intraday rebalancing required to realize continuous-time rank returns. While our proposed intraday rebalancing mechanism (Appendix D) provides a baseline, it offers substantial room for further optimization. Future work could formulate this challenge as a stochastic control problem, leveraging physics-informed neural networks [24] to solve the resulting high-dimensional partial differential equations, or alternatively as a reinforcement learning problem based on real-market data.

Beyond financial markets, the rank-based representation introduced here may generalize to other many-particle systems, such as those encountered in many-particle physics[21], biology[8], and social sciences[12].

6 Conclusion

We have introduced a novel statistical arbitrage method that leverages the robust market structure in rank space. Although rank space and name space contain the same underlying information, the significant performance improvements achieved by DNNs in rank space highlight the critical importance of domain-informed data representations in financial machine learning. In particular, our results demonstrate how appropriate transformations of the input space can dramatically enhance the learning efficiency and performance of deep models in complex, noisy environments such as many-particle systems like equity markets.

A Implementation details

A.1 Calibrating investment universe and market decomposition

Our backtesting spans January 2006 to December 2022, covering both the subprime mortgage crisis and the highly competitive post-2010 market period. On each trading day after market close, we re-calibrate the investment universe by selecting stocks that (i) rank among the top 500 in capitalization as of day t, ensuring sufficient liquidity, and (ii) have valid historical return data available for day $t+1$. This selection procedure minimizes potential selection bias to the best of our ability.

We then perform principal component analysis (PCA) on the selected returns using a 252-day lookback window to extract leading eigenvectors as market factors F_t. Specifically, we retain the top five eigenvectors (associated with the five largest eigenvalues) in name space, and the top eigenvector in rank space. Factor loadings β_t are estimated using a 60-day lookback window, from which the transformations Φ_t and residual returns ϵ_t are computed. Cumulative residual returns x_t^L are similarly evaluated over the same 60-day window and are used as inputs to either the parametric benchmark model (Appendix E) or the deep neural networks to generate portfolio weights and calculate the resulting PnL.

A.2 Deep neural networks

We delve into the specific architecture of our neural networks, illustrated in Fig. 2. Our CNN-transformer architecture harnesses the strengths of CNN in extracting local patterns and transformers in capturing long-term dependencies. The inputs of our neural networks are the trajectories of cumulative residual returns, $x_t^L \in \mathbb{R}^{N\times L}$, processed through two layer of multi-channel convolutional networks, followed by a standard transformer encoder layer that models global relationships via multi-head attention. Specifically, in the convolutionary layer,

$$
\begin{aligned}
x_t^{(1)} &= \frac{x_t^L - \mathbb{E}(x_t^L)}{\sqrt{\mathrm{Var}(x_t^L)+\epsilon}} \times \gamma^{(1)} + \beta^{(1)}, \quad y_t^{(1)} = W^{(1)} * x_t^{(1)} + b^{(1)}, \quad z_t^{(1)} = \mathrm{ReLu}(y_t^{(1)}) + x_t^{(1)};\\
x_t^{(2)} &= \frac{z_t^{(1)} - \mathbb{E}(z_t^{(1)})}{\sqrt{\mathrm{Var}(z_t^{(1)})+\epsilon}} \times \gamma^{(2)} + \beta^{(2)}, \quad y_t^{(2)} = W^{(2)} * x_t^{(2)} + b^{(2)}, \quad z_t^{(2)} = \mathrm{ReLu}(y_t^{(2)}) + x_t^{(2)}.
\end{aligned}
\tag{A.1}
$$

The superscript (1) or (2) specifies the layer number. $x_t^{(1)} \in \mathbb{R}^{N\times L}$ is the input of the first convolutional layer. $W^{(1)} \in \mathbb{R}^{D_{\text{channel}}\times 1\times D_{\text{kernel}}}$ and $W^{(2)} \in \mathbb{R}^{D_{\text{channel}}\times D_{\text{channel}}\times D_{\text{kernel}}}$ are the convolutionary kernels for the convolution operator denoted by $*$, $b^{(1,2)} \in \mathbb{R}^{D_{\text{channel}}}$ is the bias, and $y_t^{(1,2)} \in \mathbb{R}^{N\times D_{\text{channel}}\times L}$ is the output of convolutionary operator. D_{channel} is the number of channels and D_{kernel} is the size of the convolution kernel. We adopt a rectified linear unit (denoted as $\mathrm{ReLu}(\cdot)$) as our activation function. We also apply (i) instance normalization [26] with learnable parameter $\gamma^{(1,2)}$ and $\beta^{(1,2)}$ at the input of each convolution layer to accelerate the training process, and (ii) residual connection[15] to avoid vanishing gradients by directly connecting the input $x_t^{(1,2)}$ to the output $z_t^{(1,2)} \in \mathbb{R}^{N\times D_{\text{channel}}\times T}$. We choose the hyper-parameters for our neural networks as number of channels $D_{\text{channel}} = 8$ and size of the convolution kernel $D_{\text{kernel}} = 2$.

The outputs of convolutionary layers, $z_t^{(2)} \in \mathbb{R}^{N\times D_{\text{channel}}\times T}$ are subsequently fed into a standard transformer encoder layer[27]. The transformer encoder layer utilizes the multi-head attention modeled by the inner product between the famous key-query-value matrices. To elaborate,

$$
\begin{aligned}
&x_t^{\text{transformer}} = (z_t^{(2)})^T \\
&\begin{cases} Q_i &= \text{DropOut}(W_i^Q x_t^{\text{transformer}} + b_i^Q) \\ K_i &= \text{DropOut}(W_i^K x_t^{\text{transformer}} + b_i^K) \\ V_i &= \text{DropOut}(W_i^V x_t^{\text{transformer}} + b_i^V) \end{cases}, \quad i = 1, 2, ..., H \\
&\text{head}_i = \text{softmax}(\frac{Q_i K_i^T}{\sqrt{d_{\text{channel}}/H}}), \quad i = 1, 2, ..., H \\
&y_t = \text{Concat}(\text{head}_1 V_1, ..., \text{head}_H V_H) \\
&z_t = \text{LayerNorm}(x_t^{\text{transformer}} + y_t) \\
&o_t = \text{LayerNorm}(W^O z_t + b^O + y_t),
\end{aligned}
\tag{A.2}
$$

where $x_t^{\text{transformer}} \in \mathbb{R}^{N \times T \times D_{\text{channel}}}$ is the input of the transformer encoder layer. In addition, $W_i^Q \in \mathbb{R}^{(D_{\text{channel}}/H) \times D_{\text{channel}}}$, $W_i^K \in \mathbb{R}^{(D_{\text{channel}}/H) \times D_{\text{channel}}}$, and $W_i^V \in \mathbb{R}^{D_{(\text{channel}}/H) \times D_{\text{channel}}}$ are the linear weights. $b_i^Q \in \mathbb{R}^{D_{\text{channel}}/H}$, $b_i^K \in \mathbb{R}^{D_{\text{channel}}/H}$, and $b_i^V \in \mathbb{R}^{D_{\text{channel}}/H}$ are the bias. Softmax$(\cdot)$ stands for softmax function and Concat$(\cdot)$ for matrix concatenation, and $y_t \in \mathbb{R}^{N \times T \times D_{\text{channel}}}$ is the output of multi-attention layer. $W^O \in \mathbb{R}^{D_{\text{channel}} \times D_{\text{channel}}}$ and $b^O \in \mathbb{R}^{D_{\text{channel}}}$ are the linear weights and bias in the output linear layer. $o_t \in \mathbb{R}^{N \times T \times D_{\text{channel}}}$ is the output of the transformer. In addition to the residual connection similar to the convolutional layer, we also introduce the drop-out technique, denoted as Dropout$(\cdot)$, to regularize overfitting with drop-out probability p, and layer normalization[5], denoted as LayerNorm$(\cdot)$, to improve training stability. We choose the hyper-parameters for our neural networks as the number of heads $H = 4$ and the drop-out probability $p = 0.25$.

Finally, we choose the last slice along the time axis in the output of the transformer, $o_t \in \mathbb{R}^{N \times T \times D_{\text{channel}}}$, as the hidden state summarizing the information up to time t. The portfolio weights in residual space $w_t^{\epsilon|\text{NN,name/rank}}$ are calculated by a linear relationship,

$$
w_t^{\epsilon|\text{NN,name/rank}} = W^F(o_t[:, \text{-1}, :]) + b^F, \tag{A.3}
$$

where $o_t[:, \text{-1}, :] \in \mathbb{R}^{N \times D_{\text{channel}}}$ means the last slice along the second-dimension (time-axis) of o_t, and $W^F \in \mathbb{R}^{1 \times D_{\text{channel}}}$, $b^F \in \mathbb{R}$ are the parameters.

For the mean-variance optimization target in (3.10), the empirical expectation and variance are obtained over a consecutive time window of length T,

$$
\begin{aligned}
\mathbb{E}[(w_t^{R|\text{NN}})^T(r_{t+1} - r_f)] &\approx \frac{1}{T}\sum_{\alpha=1}^{T}(w_{t+\alpha}^{R|\text{NN}})^T(r_{t+\alpha+1} - r_f) \\
\text{Var}[(w_t^{R|\text{NN}})^T(r_{t+1} - r_f)] &\approx \frac{1}{T}\sum_{\alpha=1}^{T}[(w_{t+\alpha}^{R|\text{NN}})^T(r_{t+\alpha+1} - r_f) - \mathbb{E}((w_t^{R|\text{NN}})^T(r_{t+1} - r_f))]^2
\end{aligned}
\tag{A.4}
$$

We choose the risk-aversion factor $\gamma = 2$ and length of time window $T = 24$ days.

The neural networks are trained in two steps. The first step aims at optimizing the hyper-parameters of neural network. Specifically, to evaluate the portfolio weights from the trading day t to day $t + 252$, we utilize the data from $t - 1000$ to $t - 60$ as the training data set and from $t - 59$ to $t - 1$ as the validation data set, from which we determine hyper-parameters from the converged mean-variance target in the training curve (Fig. 7). The hyperparameters of our optimized neural network architecture are elaborated in Appendix A.2.

The second step aims at increasing the updating frequency of parameters in the neural networks from annually to quarterly while keeping hyper-parameters fixed. More explicitly, to evaluate the portfolio weights from the trading day t to day $t + 63$, we use the data from trading day $t - 500$ to day $t - 1$ as the training data set. Empirically, increasing the updating frequency is significantly beneficial to the performance of neural network, due to the non-stationarity in financial data. The training tasks utilize PyTorch 2.2.0, and are parallelized on a workstation with a CPU from AMD Ryzen Threadripper Pro 5955 WX and two GPUs from Nvidia GeForce RTX 4090. Each neural network training iteration takes approximately two hours to converge, resulting in roughly 130 computational hours for backtesting portfolio performance from 2007 to 2022, assuming quarterly neural network retraining. In our rank-space statistical arbitrage strategy, the daily portfolio weights in rank space are precomputed by forward propagation of the neural network before market opening, and remain fixed during the trading day. To handle rank changes, the intraday rebalancing of portfolio weights from rank space to name space occurs every 225 minutes.

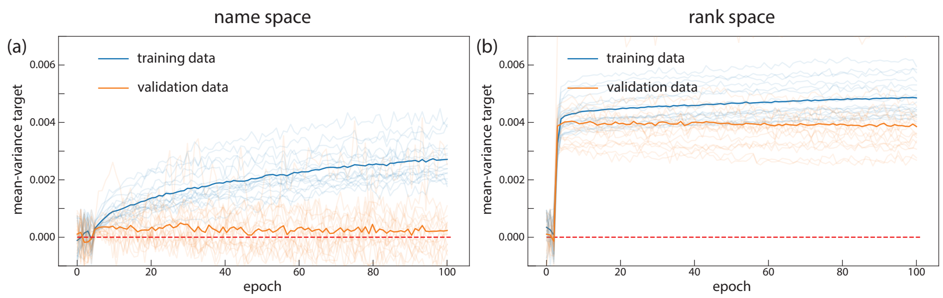

Figure 7: Training curves of neural networks. **(a, b)** We show the mean-variance target as a function of training epochs for neural networks as specified by (3.10) in name space (a) and in rank space (b). The training curves originate from the phase I training process focusing on hyperparameter tuning. To evaluate the out-of-sample portfolio weights from trading day t to $t + 252$, we use the training data from day $t - 1000$ to day $t - 60$ and validation data from day $t - 59$ to $t - 1$ as the validation data. The neural networks are re-trained annually with random initialization, yielding approximately 17 training curves from 2006 to 2022. The transparent lines show individual training curves with their average represented by the opaque lines. The neural networks in rank space are more efficiently trained than those in name space.

A.3 Algorithms

We provide implementation details in the form of pseudocode. Algorithm 1 performs market decomposition in both name and rank space, as discussed in section 3.1. Algorithm 2 calculates the portfolio weights using the parametric model (section E), while algorithm 3 does so via neural networks (section 3.2 and A.2). Algorithm 4 handles the conversion of portfolio weights between name space and rank space (section D). Algorithm 5 computes the summary statistics of portfolio performance reported in 1 and 2. Finally, algorithm 6 and 7 integrate the proposed algorithmic components above to implement the complete statistical arbitrage strategy in name and rank space, respectively.

Algorithm 1 Market decomposition (PCA) [Fig. 1, panel(c1, c2)]

Input: $r_t, r_{f,t}, K$
Output: ϵ_t, Φ_t

Function market_decomposition($r_t, r_{f,t}, K$):
- Principal component analysis: $r_t - r_{f,t} = U\Sigma V^T$
- $F_t \leftarrow (v_1, v_2, ..., v_K)$, where v_k is the k-th column of V^T
- Calculate ω_t by solving $F_t = \omega_t(r_t - r_f)$
- Calculate β_t as the coefficient of the linear regression $r_t - r_f \sim F_t$
- $\Phi_t \leftarrow I - \beta_t\omega_t$
- $\epsilon_t \leftarrow \Phi_t(r_t - r_{f,t})$
- **return** ϵ_t, Φ_t

// Input:
  // r_t: return in name space or transformed return in rank space.
  // $r_{f,t}$: risk-free rate at the end of trading day t.
  // K: number of market factors, predetermined by analyzing eigenvalue
  // spectrum of the correlation matrix.
// Output:
  // ϵ_t: residual returns in name space or rank space.
  // Φ_t: transformation between residual space and equity space
  // ((3.1) for name space and (3.6) for rank space).
// Note:
  // The algorithm realizes the formulation in section 2.1.
  // Factors F_t and ω_t are calculated on a 252-day look-back window.
  // Loadings β_t are calculated on a 60-day look-back window.
  // F_t, ω_t, and β_t are updated daily.
  // $K = 5$ for name space and $K = 1$ for rank space based on empirical
  // eigenvalue spectrum of the correlation matrix (Fig. 3(c,d)).

Algorithm 2 Portfolio weights by parametric model [Fig. 1, panel(d1, e1)]

Input: ϵ_t, Φ_t
Output: $w_t^{R|\mathrm{OU}}$

Function portfolio_weights_by_parametric_model(ϵ_t, Φ_t)
  Calculate x_t^L by (3.9)
  Estimate $\tau, \mu, \sigma, x_t, R^2$ by fitting x_t^L to an OU process ((E.1))
  Calculate $w_t^{\epsilon|\mathrm{OU}}$ by (E.4)
  $w_t^{R|\mathrm{OU}} \leftarrow \Phi_t^T w_t^{\epsilon|\mathrm{OU}}$((E.7))
  return $w_t^{R|\mathrm{OU}} / ||w_t^{R|\mathrm{OU}}||_1$

```
// Input:
// ϵ_t:  residual returns calculated from Algorithm 1
// Φ_t:  transformation matrix between equity space and residual space
from
     // Algorithm 1
// Output:
// w_t^{R|OU}:  l_1-normalized portfolio weights by parametric model.
        // For name space, it stands for the portfolio weights on stocks.
        // For rank space, it corresponds to the portfolio weights on
        // artificial financial instruments that realize rank returns
        // defined in (3.5).
// Note:
    // The algorithm realizes the formulation in section 2.2.1.
    // τ, μ, σ are fitting parameters of OU process.
    // Risk control by ignoring τ > 30 days ((E.4)).
```

Algorithm 3 Portfolio weights by neural networks [Fig. 1, panel(d2, e2)]

Input: ϵ_t, Φ_t
Output: $w_t^{R|\mathrm{NN}}$

Function portfolio_weights_by_neural_networks(ϵ_t, Φ_t)
  Calculate x_t^L by (3.9)
  Train neural network in-sample for mean-variance optimization ((3.10))
  Calculate $w_t^{\epsilon|\mathrm{NN}}$ out-of-sample from trained neural network
  return $w_t^{R|\mathrm{NN}} / ||w_t^{R|\mathrm{NN}}||_1$

```
// Input:
// ϵ_t:  residual returns calculated from Algorithm 1.
// Φ_t:  transformation matrix between equity space and residual space
from
     // Algorithm 1.
// Output:
// w_t^{R|NN}:  l_1-normalized portfolio weights by parametric model.
        // For name space, it stands for the portfolio weights on stocks.
        // For rank space, it corresponds to the portfolio weights on
        // artificial financial instruments that realize rank returns.
        // defined in (3.5).
// Note:
    // The algorithm realizes the formulation in section 2.2.2.
    // No pre-screening on trading opportunities x_t^L applied.
    // Neural networks internally prioritize various trading opportunities
        // and manage risk (Fig. 6).
    // The mean-variance target is evaluated on a 24-day window.
```

Algorithm 4 intraday rebalancing [Fig. 1, panel (f3)]

Input: $w_t^R, r_{f,t}, \mathcal{T}$
    $c_{t+\tau}, \quad t = 1, 2, ..., T$ (days), $\tau = 1, 2, ..., N$ (minutes)
Output: $V_t, \quad t = 1, 2, ..., T+1$

Function intraday_rebalancing($w_t^R, r_{f,t}, \mathcal{T}, c_{t+\tau}$)
  $t \leftarrow 0, V_t \leftarrow 1, w^{\text{prev}} \leftarrow 0$
  While $t \leq T$ ▷ T is daily time tick
    $w_{(k),t}^{\text{rank}} \leftarrow w_{(k),t}^R, \quad k = 1, 2, ..., N$
    $w_{\mathcal{I}_{(k),t},t}^{\text{name}} \leftarrow w_{(k),t}^R, \quad k = 1, 2, ..., N$ ▷ $\mathcal{I}_{(k),t}$ maps from rank to name
    $V_t \leftarrow V_t - \sum_i w_{i,t}^{\text{name}} - 0.0002 \times ||w_t^{\text{name}} - w^{\text{prev}}||_1$
    $\tau \leftarrow 0$
    While $t + \tau < (t+1)$ ▷ τ is intraday time tick
      $w_{(k),t+\tau}^{\text{rank}} \leftarrow w_{(k),t+\tau-1}^{\text{rank}} \times \frac{c_{(k),t+\tau}}{c_{(k),t+\tau-1}}, k = 1, 2, ..., N$ ((3.12))
      $w_{i,t+\tau}^{\text{name}} \leftarrow w_{i,t+\tau-1}^{\text{name}} \times \frac{c_{i,t+\tau}}{c_{i,t+\tau-1}}, i = 1, 2, ..., N$ ((3.13))
      ▷ $t + \tau - 1$ and $t + \tau$ are adjacent intraday timestamps
      if $\tau \% \mathcal{T} == 0$ **or** end of the trading day ▷ rebalancing point
        Calculate cost($t + \tau, w_{(k),t}^{\text{rank}}$) by (3.14)
        $V_t \leftarrow V_t - \text{cost}(t + \tau, w_{(k),t}^{\text{rank}})$
        $w_{\mathcal{I}_{(k),t+\tau},t+\tau}^{\text{name}} \leftarrow w_{(k),t+\tau}^{\text{rank}}, \quad k = 1, 2, ..., N$
      $\tau \leftarrow \tau + 1$
    $V_{t+1} \leftarrow (1 + r_{f,t+1})V_t + \sum_i w_{i,t+\tau}^{\text{name}}$
    $w^{\text{prev}} \leftarrow w_{t+\tau}^{\text{name}}$
    $t \leftarrow t + 1$
  return $V_t, \quad t = 1, 2, ..., T+1$

// Input:
  // w_t^R: the l_1-normalized portfolio weights from either parametric
    // model (Algorithm 2) or neural networks (Algorithm 3).
  // $r_{f,t}$: risk-free rate during the trading day t.
  // $\mathcal{T}$: rebalance interval.
  // $c_{t+\tau}$: the capitalization processes in name space and rank space at
    // 1-minute resolution throughout the trading day t.
    // t is the time tick at daily level.
    // τ is the time tick at minute level.
// Output:
  // V_t: the value process of the portfolio (PnL) with weights w_t^R.
// Note:
  // The algorithm realizes the formulation in section 2.3.1.
  // In essence, it converts portfolio weights from rank to name
    // at $\mathcal{T}$ minutes interval.

Algorithm 5 portfolio metric [Fig. 1, panel (g)]

Input: $V_t, \quad t = 1, 2, ..., N$
Output: $r_{\text{annual}}, \sigma_{\text{annual}}, \text{SR}_{\text{annual}}$

Function portfolio_metric(V_t, r_f)
  For year **in** years_in_backtesting:
    Locate all $t_1 \leq t_2 \leq ... \leq t_N$ in year
    $r_{t_i} = V_{t_i}/V_{t_{i-1}} - 1, \quad i = 1, 2, ..., N$
    $r_{\text{annual}} \leftarrow (\prod_{i=1}^{N}(1 + r_{t_i}))^{252/N} - 1$
    $\sigma_{\text{annual}} \leftarrow \sqrt{252} \times \text{std}(\{r_{t_i}\}_{i=1}^{N})$ ▷ std is the standard deviation
    $\text{SR}_{\text{annual}} \leftarrow (r_{\text{annual}} - r_{f,\text{annual}})/\sigma_{\text{annual}}$ ▷ $r_{f,\text{annual}}$ is annualized risk-free rate
  return $r_{\text{annual}}, \sigma_{\text{annual}}, \text{SR}_{\text{annual}}$ for all backtesting years

// Input:
  // V_t: the value process (PnL) of the portfolio with weights w_t^R.
  // $r_{f,t}$: the risk-free rate at the end of trading day t.
// Output:
  // The algorithm realizes the formulation in section 2.4.
  // r_{annual}: the annualized return for all backtesting years.
  // σ_{annual}: the annualized volatility for all backtesting years.
  // $\text{SR}_{\text{annual}}$: the Sharpe ratio for all backtesting years.

Algorithm 6 (Integrated) Statistical arbitrage in name space

Input: r_t, r_f, K
Called algorithm: Algorithm 1, Algorithm 2, Algorithm 3, Algorithm 5
Output: w_t^R, V_t, $\text{SR}_{\text{annual}}$
Function statistical_arbitrage_in_name_space($r_t, r_{f,t}, K$)
  $\epsilon_t, \Phi_t \leftarrow$ market_decomposition($r_t, r_{f,t}, K$) from Algorithm 1
  if parametric model
    $w_t^{R|\text{OU}} \leftarrow$ portfolio_weights_by_parametric_model(ϵ_t, Φ_t) from Algorithm 2;
  if neural networks
    $w_t^{R|\text{NN}} \leftarrow$ portfolio_weights_by_neural_network(ϵ_t, Φ_t) from Algorithm 3
  Calculate PnL V_t by (3.15)
  $r_{\text{annual}}, \sigma_{\text{annual}}, \text{SR}_{\text{annual}} \leftarrow$ portfolio_metric($V_t, r_{f,t}$) from Algorithm 5
  return w_t^R, V_t, $\text{SR}_{\text{annual}}$

// Input:
  // r_t: dividend-adjusted daily return in name space.
  // $r_{f,t}$: risk-free rate at the end of trading day t.
  // K: number of market factors, predetermined by analyzing eigenvalue
  // spectrum of the correlation matrix.
// Output:
  // w_t^R: the l_1-normalized portfolio weights on stock.
  // V_t: the value process (PnL) of the portfolio with weights w_t^R.
  // r_{annual}, σ_{annual}, $\text{SR}_{\text{annual}}$: annualized return, volatility, and Sharpe
  // ratio.

Algorithm 7 (Integrated) Statistical arbitrage in rank space

Input: $c_t, c_{t+\tau}, r_f, K$
Called algorithm: Algorithm 1, Algorithm 2, Algorithm 3, Algorithm 4, Algorithm 5
Output: w_t^R, V_t, r_{annual}, σ_{annual}, $\text{SR}_{\text{annual}}$

Function statistical_arbitrage_in_name_space($c_t, r_{f,t}, K$)
  Calculate $\tilde{r}_t$ by (3.5)
  $\tilde{\epsilon}_t, \tilde{\Phi}_t \leftarrow$ market_decomposition($\tilde{r}_t, r_{f,t}, K$) from Algorithm 1
  if parametric model
    $w_t^{R|\text{OU}} \leftarrow$ portfolio_weights_by_parametric_model(ϵ_t, Φ_t) from Algorithm 2;
  if neural networks
    $w_t^{R|\text{NN}} \leftarrow$ portfolio_weights_by_neural_network(ϵ_t, Φ_t) from Algorithm 3
  $V_t \leftarrow$ intraday_rebalancing($w_t^{R|\text{NN}}, r_{f,t}, \mathcal{T}, c_{t+\tau}$) from Algorithm 4
  r_{annual}, σ_{annual}, $\text{SR}_{\text{annual}} \leftarrow$ portfolio_metric($V_t, r_{f,t}$) from Algorithm 5
  return w_t^R, V_t, r_{annual}, σ_{annual}, $\text{SR}_{\text{annual}}$

`// Input:`
  `//` c_t`: capitalizations at the end of trading day` t`.`
  `//` $c_{t+\tau}$`: capitalization process at 1-minute resolution throughout the`
    `// trading day` t`.` t `is the time tick at daily level.`
    `//` τ `is the time tick at intraday level.`
  `//` $r_{f,t}$`: risk-free rate at the end of trading day` t`.`
    `//` K`: number of market factors, predetermined by analyzing`
`eigenvalue`
    `// spectrum of the correlation matrix.`
`// Output:`
  `//` w_t^R`: the` l_1`-normalized portfolio weights on artificial financial`
    `// instruments that realize` $\tilde{r}_t$`.`
  `//` V_t`: the value process (PnL) of the portfolio with weights` w_t^R`.`
  `//` r_{annual}`,` σ_{annual}`,` $\text{SR}_{\text{annual}}$`: annualized return, volatility, and Sharpe`
    `// ratio.`

B Hybrid-Atlas model

In this section, we introduce a hybrid-Atlas model that motivates the definition of return in (3.5) and hints the qualitative difference between name space and rank space[7, 6, 18].

We study an equity market that consists of n stocks with capitalizations $C(t) = (C_1(t), \ldots, C_n(t))$, where $C_i(t)$ represents the capitalization at time t of the asset with name i. We assume that the log-capitalizations $Y_i(t) := \log C_i(t)$, $i = 1, \ldots, n$, satisfy the system of stochastic differential equations:

$$\begin{aligned} dY_i(t) &= \left(g_{\mathcal{R}_{i,t}} + \gamma_i + \gamma\right) dt + \sum_{j=1}^{n} \rho_{i,j}\, dW_j(t) \\ &\quad + \sigma_{\mathcal{R}_{i,t}} Y_i(t)\, dW_i(t), \quad Y_i(0) = y_i, \quad 0 \le t < \infty \end{aligned} \tag{B.1}$$

with given initial condition $y = (y_1, \ldots, y_n)'$. We assume that $\gamma = 0$ and the system satisfies the stability condition

$$\sum_{k=1}^{n} g_k + \sum_{i=1}^{n} \gamma_i = 0. \tag{B.2}$$

Define the log-capitalization process in rank space, $Z_k(t) = Y_{\mathcal{I}_{k,t}}(t)$. Therefore, the rank return defined in (3.5) can be viewed as the discrete counterpart of $dZ_k(t)$.

Theorem B.3 highlights the qualitative difference between residual rank space and name space beyond linear transformation.

Lemma B.1. *For continuous semimartingales $Y_1, Y_2, ..., Y_n$, the rank process $Z_1, Z_2, ..., Z_n$ are continuous semimartingales, and we have*

$$\sum_{k=1}^{n} L_t(Z_k) = \sum_{i=1}^{n} L_t(Y_i), \quad \forall t > 0, \tag{B.3}$$

where $L_t(Y) = \frac{1}{2\epsilon} \int_0^t \mathbf{1}_{\{-\epsilon < Y_s < \epsilon\}} dY_s$ is the local time accumulated at origin.

Proof. The proof follows [6] theorem 2.2. □

Lemma B.2. *For continuous semimartingales $Y_1, Y_2, ..., Y_n$ and their rank process $Z_1, Z_2, ..., Z_n$, we have*

$$\begin{aligned} dZ_k(t) &= \sum_{i=1}^{n} (N_k(t))^{-1} \mathbf{1}_{\{Z_k(t) = X_i(t)\}}\, dX_i(t) \\ &\quad + \sum_{j=k+1}^{n} (N_k(t))^{-1}\, dL_t(Z_k - Z_j) - \sum_{j=1}^{k-1} (N_k(t))^{-1}\, dL_t(Z_j - Z_k). \end{aligned} \tag{B.4}$$

, where $S_t(k) := \{i : Y_i(t) = Z_k(t)\}$ and $N_k(t)$ is the cardinality of $S_t(k)$.

Proof. Our proof follows [6] theorem 2.3 closely.

Define

$$U = \{u(\cdot) : \mathbb{R}^+ \times \{1, \ldots, n\} \to \{1, \ldots, n\}, Z_k(t) = Y_{u_t(k)}(t), \forall t > 0,\ k = 1, \ldots, n\}. \tag{B.5}$$

For any $\mathbf{j} \in J$, we define $u^{\mathbf{j}}$ of U as

$$u_t^{\mathbf{j}}(k) := \text{the } j_{N_t(k)}\text{-th smallest element of } S_t(k). \tag{B.6}$$

In other words, suppose that at time t precisely m of the processes $X_1, \ldots, X_n$ have rank k, denoted as $X_{i_1}, \ldots, X_{i_m}$. Then, $u_t^{\mathbf{j}}(k)$ is the j_m-th smallest among the indices $i_1, \ldots, i_m$.

For $u \in U$, we have

$$
\begin{aligned}
Z_k(t) - Z_k(0) &= \sum_{i=1}^{n} \int_0^t \mathbf{1}_{\{u_s(k)=i\}}\, dY_i(s) + \sum_{i=1}^{n} \int_0^t \mathbf{1}_{\{u_s(k)=i\}}\, d\left(Z_k(s) - Y_i(s)\right) \\
&= \frac{1}{n!} \sum_{i=1}^{n} \int_0^t \sum_{j\in J} \mathbf{1}_{\{u_s^{\mathrm{j}}(k)=i\}}\, dY_i(s) + \frac{1}{n!} \sum_{i=1}^{n} \int_0^t \sum_{j\in J} \mathbf{1}_{\{u_s^{\mathrm{j}}(k)=i\}}\, d\left(X_{(k)}(s) - X_i(s)\right). \\
&= \sum_{i=1}^{n} \int_0^t (N_s(k))^{-1} \mathbf{1}_{\{Z_k(s)=Y_i(s)\}}\, dY_i(s) + \sum_{i=1}^{n} \int_0^t (N_s(k))^{-1} \mathbf{1}_{\{Z_k(s)=Y_i(s)\}}\, d\left(Z_k(s) - Y_i(s)\right) \\
&= \sum_{i=1}^{n} \int_0^t (N_s(k))^{-1} \mathbf{1}_{\{Z_k(s)=Y_i(s)\}}\, dY_i(s) + \sum_{i=1}^{n} \int_0^t (N_s(k))^{-1}\, dL_s\left((Z_k - Y_i)^{+}\right) \\
&\quad - \sum_{i=1}^{n} \int_0^t (N_s(k))^{-1}\, dL_s\left((Z_k - Y_i)^{-}\right).
\end{aligned}
\tag{B.7}
$$

, where $(\cdot)^{+} = \max(\cdot, 0)$ and $(\cdot)^{-} = \min(\cdot, 0)$. In the second equality, we replace u by u^j for $j \in J$, then summing over all j. In the third equality, we use $\sum_{j\in J} \mathbf{1}_{\{u_s^{\mathrm{j}}(k)=i\}} = \frac{n!}{N_s(k)} \mathbf{1}_{\{X_{(k)}(s)=X_i(s)\}}$. In the fourth equality, we use lemma B.1 and

$$
\begin{aligned}
&\sum_{i=1}^{n} \int_0^t (N_s(k))^{-1} \mathbf{1}_{\{Z_k(s)=Y_i(s)\}}\, d\left(Z_k(s) - Y_i(s)\right) \\
&= \sum_{i=1}^{n} \int_0^t (N_s(k))^{-1} \mathbf{1}_{\{Z_k(s)=Y_i(s)\}}\, d\left((Z_k(s) - Y_i(s))^{+}\right) \\
&\quad - \sum_{i=1}^{n} \int_0^t (N_s(k))^{-1} \mathbf{1}_{\{Z_k(s)=Y_i(s)\}}\, d\left((Z_k(s) - Y_i(s))^{-}\right) \\
&= \sum_{i=1}^{n} \int_0^t (N_s(k))^{-1}\, dL_s\left((Z_k - Y_i)^{+}\right) - \sum_{i=1}^{n} \int_0^t (N_s(k))^{-1}\, dL_s\left((Z_k - Y_i)^{-}\right).
\end{aligned}
\tag{B.8}
$$

□

Theorem B.3. *The log-capitalization process in rank space Z_k satisfies*

$$
\begin{aligned}
dZ_k(t) &= \left(g_{\mathcal{R}_{i,t}} + \gamma_i + \gamma\right) dt + \sum_{j=1}^{n} \rho_{i,j}\, dW_j(t) + \sigma_{\mathcal{R}_{i,t}} Y_i(t)\, dW_i(t), dY_{\mathcal{I}_{k,t}}(t) \\
&\quad + \frac{1}{2}(d\Lambda^{k,k+1}(t) - d\Lambda^{k-1,k}(t))
\end{aligned}
\tag{B.9}
$$

, where $\Lambda^{k,k+1}$ is the local time accumulated at the origin by the non-negative semimartingale $Z_k(t) - Z_{k+1}(t)$ defined as

$$
\Lambda^{k,k+1}(t) = \lim_{\varepsilon \downarrow 0} \frac{1}{2\varepsilon} \int_0^t \mathbf{1}_{\{-\varepsilon < Z_k(s) - Z_{k+1}(s) < \varepsilon\}} d(Z_k(s) - Z_{k+1}(s)) \tag{B.10}
$$

Proof. From the lemmas above, we have

$$
\begin{aligned}
dZ_k(t) &= \left(g_{\mathcal{R}_{i,t}} + \gamma_i + \gamma\right) dt + \sum_{j=1}^{n} \rho_{i,j}\, dW_j(t) + \sigma_{\mathcal{R}_{i,t}} Y_i(t)\, dW_i(t), dY_{\mathcal{I}_{k,t}}(t) \\
&\quad + (N_k(t))^{-1} \left[\sum_{\ell=k+1}^{n} d\Lambda^{k,\ell}(t) - \sum_{\ell=1}^{k-1} d\Lambda^{\ell,k}(t) \right] \\
&= \left(g_{\mathcal{R}_{i,t}} + \gamma_i + \gamma\right) dt + \sum_{j=1}^{n} \rho_{i,j}\, dW_j(t) + \sigma_{\mathcal{R}_{i,t}} Y_i(t)\, dW_i(t), dY_{\mathcal{I}_{k,t}}(t) \\
&\quad + \frac{1}{2}(d\Lambda^{k,k+1}(t) - d\Lambda^{k-1,k}(t))
\end{aligned}
\tag{B.11}
$$

, where the second equality follows from [18] lemma 1 that the local times $\Lambda^{k,l}$ generated by triple or higher-order collisions are identically equal to zero. In other words, $\Lambda^{k,l} = 0$ for $|k-l| \geq 2, 1 \leq k, l, \leq n$. □

C Market neutrality

The statistical arbitrage portfolios need to satisfy the market neutrality constraint, $w^T\beta = 0$, so that the return of the portfolio is independent of market factors

$$w^T(r_t - r_f) = w^T(\beta_t F_t + \epsilon_t) = w^T\epsilon_t. \tag{C.1}$$

The following theorem proves that the market neutrality of the portfolio constructed in section 3.1 satisfies the constraints.

Theorem C.1. *If the portfolio weights satisfies Eq. 3.4 or Eq. 3.8, it is market neutral.*

Proof. We denote the return matrix $R_t = (r_{t-T+1}, r_{t-T+2}, ..., r_t) \in \mathbb{R}^{N\times T}$. Assume singular value decomposition of $R_t - R_f$,

$$R_t - R_f = U\Sigma V^T \tag{C.2}$$

, where $R_f \in \mathbb{R}^{1\times T}$ is the risk-free rate, $U \in \mathbb{R}^{N\times N}$, $\Sigma \in \mathbb{R}^{N\times T}$, and $V^T \in \mathbb{R}^{T\times T}$. Then, the factors and loadings in Eq. 3.1 and ω_t in Eq. 3.2 becomes

$$F_t = \begin{pmatrix} v_1^T \\ v_2^T \\ ... \\ v_K^T \end{pmatrix}, \quad \beta_t = (u_1, u_2, ..., u_K)\begin{pmatrix} \sigma_1 & & \\ & ... & \\ & & \sigma_K \end{pmatrix}, \quad \omega_t = \begin{pmatrix} \sigma_1^{-1} & & \\ & ... & \\ & & \sigma_K^{-1} \end{pmatrix}\begin{pmatrix} u_1^T \\ u_2^T \\ ... \\ u_K^T \end{pmatrix} \tag{C.3}$$

, where u_i and v_i are the i-th column of matrix U and V. Then, due to the orthogonality between U and V,

$$\Phi_t\beta_t = (I - \beta_t\omega_t)\beta_t = \beta_t - \beta_t(\omega_t\beta_t) = \beta_t - \beta_t = 0, \tag{C.4}$$

. Therefore,

$$(w_t^R)^T\beta = (w_t^\epsilon)^T\Phi_t\beta = 0 \tag{C.5}$$

$\square$

D Intraday rebalancing

The portfolio weights calculated in the rank space are assigned to artificial financial instruments that yield rank returns in continuous-time limits defined in (3.5). To make the constructed portfolio practically implementable, it is necessary to convert these portfolio weights into stock-based portfolios in name space.

A naive approach is to assign the portfolio weights with correspondence between ranks and names at the end of each trading day and hold the portfolio throughout the following trading day,

$$w_{i,t} = \sum_{k=1}^{N} w_{(k),t} \mathbf{1}_{\{\mathcal{R}_{i,t}=k\}}, \qquad i = 1, 2, ..., N. \tag{D.1}$$

Unfortunately, this straightforward conversion will not retain the advantages of statistical arbitrage in rank space, because the performance of the derived portfolio will essentially still depend on returns in name space rather than the rank returns in continuous-time limit. As indicated by (3.5), the returns in name and rank space start to diverge in the event of rank switching that frequently occurs for most ranks at an intraday frequency. It indicates that an effective conversion strategy must appropriately respond to the rank-switching events.

Consequently, we propose an intraday rebalancing mechanism in section 3.3. This mechanism performs conversion from rank space to name space at a frequency that matches rank-switching events, even though it results in higher transaction costs due to more frequent trading. In the following, we carry out an in-depth analysis in a two-stock system to emphasize the crucial role of rank switching and the rebalancing interval in determining the cost of conversion.

D.1 Portfolio rebalancing through rank switching of two stocks

To elucidate the pivotal role of rank switching in our intraday rebalancing strategy, we examine a two-stock system depicted in Fig. 8, where the two capitalization processes $c_{t,1}$ and $c_{t,2}$ maintain their ranks during the rebalancing interval $((k-1)\mathcal{T}, k\mathcal{T}], k \in \mathbb{N}$ and swap their ranks during $(k\mathcal{T}, (k+1)\mathcal{T}], k \in \mathbb{N}$ (panels (a1-a7)). The red lines in panels (b1-b7) and green lines in panels (c1-c7) show the dollar portfolio weight on the stock that occupies k-th rank in capitalization, $w_{(k),\tau}, k = 1, 2$. The orange lines in panels (b1-b7) and blue lines in panels (c1-c7) show the dollar portfolio weight on the stock that has i-th name index, $w_{i,\tau}, i = 1, 2$. We further calculate and present in panels (d1-d7) the divergence between the total dollar portfolio weights in rank space and the total dollar portfolio weights in name space, defined as $w_{(1),t} + w_{(2),t} - w_{1,t} - w_{2,t}$. Panels (e1-e7) shows the cumulative cost from the bid-ask spread arising from the active trading at the rebalancing point. We highlight several representative timestamps elaborated below.

(i) $t = (k-1)\mathcal{T}^+$ in panels (a1-e1): We invest $w_{(2),t}$ on stock 1 and $w_{(1),t}$ on stock 2 since $c_{1,t} < c_{2,t}$. Therefore, $w_{(1),t} = w_{2,t}$ and $w_{(2),t} = w_{1,t}$;

(ii) $t = k\mathcal{T}$ in panel (a2-e2): The processes evolve towards the rebalancing point $k\mathcal{T}$. The relationship $w_{(1),t} = w_{2,t}, w_{(2),t} = w_{1,t}$ maintains because there is no rank-swapping between stock 1 and stock 2;

(iii) $t = k\mathcal{T}^+$ in panels (a3-e3): At the rebalancing point, no active trading is needed as $w_{(1),t} = w_{2,t}, w_{(2),t} = w_{1,t}$ for $i = 1, 2$, and therefore no divergence (latency cost) or cost from bid-ask spread incurred;

(iv) $k\mathcal{T} < t \leq (k+1)\mathcal{T}$ in panels (a4-e4, a5-e5): The processes evolve towards the rebalancing point $(k+1)\mathcal{T}$. However, because of the rank switch in capitalization between stock 1 and 2 during the interval, the dollar-valued portfolio for rank and the dollar-valued portfolio for name start diverging, i.e. $w_{(1),t} \neq w_{2,t}, w_{(2),t} \neq w_{1,t}$ and reaches a maximum at the next rebalancing point $(k+1)\mathcal{T}$ (panel (d4, d5));

(v) $t = (k+1)\mathcal{T}^+$ in panels (a6-e6): We carry out active trading to rebalancing the portfolio such that $w_{(1),t} = w_{1,t}, w_{(2),t} = w_{2,t}$. This requires cash reserve to compensate (i) divergence $w_{(1),t} + w_{(2),t} - w_{1,t} - w_{2,t}$ (panel (d6)), and (ii) cost from the bid-ask spread (e6);

(vi) $t > (k+1)\mathcal{T}^+$ in panels (a7-e7): the system continues to evolve with $w_{(1),t} = w_{1,t}, w_{(2),t} = w_{2,t}$, and the divergence becomes zero.

From the detailed analysis above, the need for active trading stems from their rank switching during the balance interval. Furthermore, the latency cost is tied to the divergence of total dollar portfolio weights between rank space and name space, $w^{\text{rank}}_{(1),t} + w^{\text{rank}}_{(2),t} - w^{\text{name}}_{1,t} - w^{\text{name}}_{2,t}$. This divergence increases with the interval between the time for rank switching and the time for the subsequent rebalancing point, suggesting that decreasing rebalancing intervals $\mathcal{T}$ might reduce the risk of large transaction costs by minimizing latency costs.

However, the situation becomes more complex when considering the fluctuating nature of the capitalization process. In the scenario where two adjacent capitalization processes frequently switch ranks, as shown in Fig. 9, trading too frequently in response to the instantaneous rank changes can incur substantial, yet unnecessary costs from bid-ask spread. To illustrate this, we present a similar two-particle system where fluctuating capitalization processes cross their paths (Fig. 9(a1-a3)). We calculate dollar portfolio weights in name space and rank space according to the aforementioned intraday rebalancing strategy, and analyze the divergence of total dollar portfolio weights between rank space and name space (Fig. 9(b1-b3)), the cumulative latency costs (Fig. 9(c1-c3)), the cumulative costs from bid-ask spread (Fig. 14(d1-d3)), and the cumulative transaction cost (Fig. 9(e1-e3)). We consider three scenarios under large, medium, and small rebalancing intervals. Remarkably, our findings underscore a return-risk trade-off: frequent trading (short rebalancing interval) yields lower divergence and hence lower risk but incurs higher costs from the bid-ask spread, whereas less frequent trading (large rebalancing interval) results in higher divergence and risk but lower bid-ask costs. Thus, selecting an appropriate rebalancing interval is crucial for minimizing overall transaction costs by balancing between latency costs and costs from bid-ask spread. Indeed, we observe a strong dependence on the profit and loss (PnL) with different intraday rebalance intervals in our empirical analysis (Fig. 14).

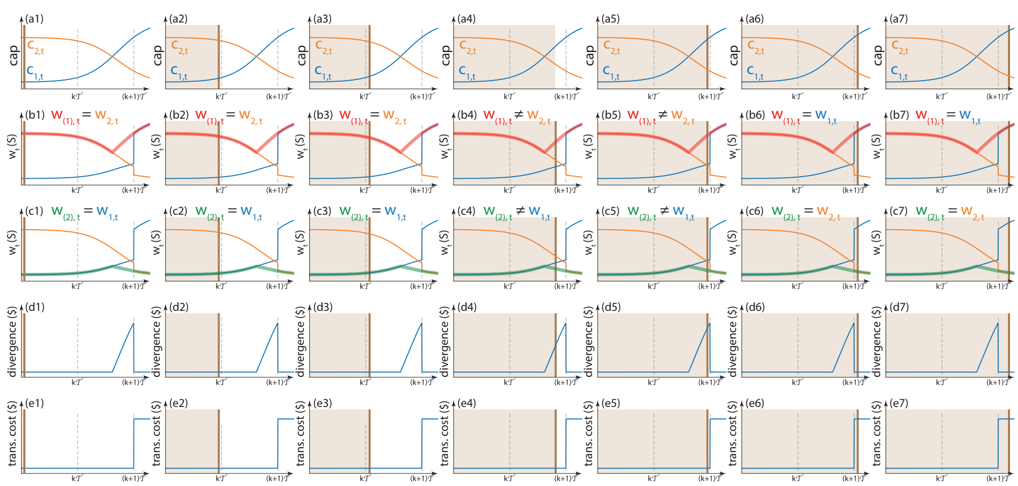

Figure 8: **Schematic for the intraday rebalancing through rank switching of two stocks.** Here, we examine a two-stock system and highlight the critical role of the rank switching. We consider two capitalization processes $c_{t,1}$ and $c_{t,2}$ that maintain their ranks during the rebalancing interval $((k-1)\mathcal{T}, k\mathcal{T}]$ and switch their ranks during $(k\mathcal{T}, (k+1)\mathcal{T}]$ (panel (a1-a7)). The red lines in panel (b1-b7) and green lines in panel (c1-c7) show the dollar-valued portfolio in rank space $w_{(1),\tau}$ and $w_{(2),\tau}$ respectively, where $w_{(k),t}$ denotes the dollar portfolio weights on the stock that occupies k-th rank in capitalization. The orange line in panel (b1-b7) and blue line in panel (c1-c7) show the dollar-valued portfolio in name space $w_{1,\tau}$ and $w_{2,\tau}$ respectively, where $w_{i,\tau}$ denotes the dollar portfolio weights on the stock that had i-th name index. We further calculate and present in panel (d1-d7) the divergence between the total dollar portfolio weights in rank space and in name space, defined as $w_{(1),t} + w_{(2),t} - w_{1,t} - w_{2,t}$. Panel (e1-e7) shows the cumulative cost from bid-ask spread arising from the active trading at the rebalancing point.

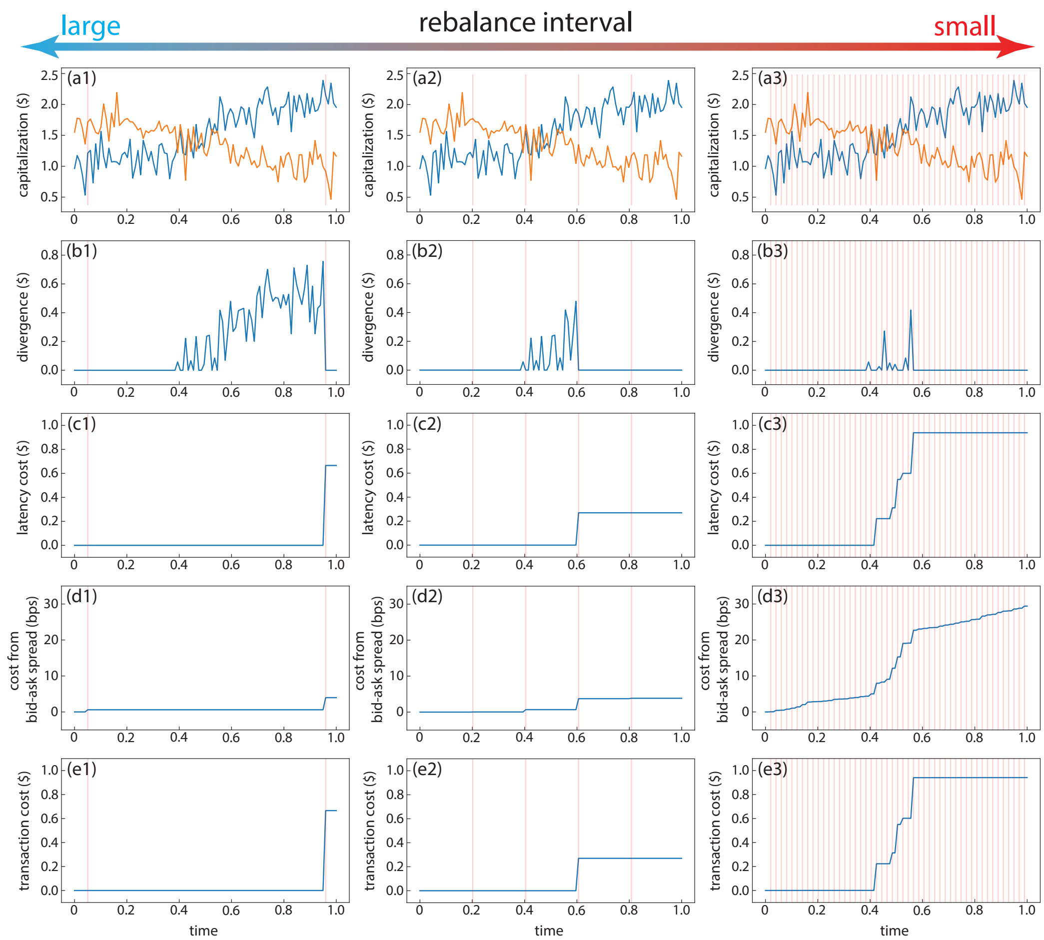

Figure 9: **Transaction cost dependence on the rebalancing interval.** This figure illustrates how transaction costs incurred by intraday rebalancing are influenced by different rebalancing intervals: large (panel (a1-e1)), medium (panel (a2-e2)), and small (panel (a1-a3)). The red vertical lines mark the rebalancing points across all panels. **(a1-a3)** The capitalization processes from two stocks. **(b1-b3)** The divergence between the total dollar portfolio weights in rank space and name space, $w_{(1),t} + w_{(2),t} - w_{1,t} + w_{2,t}$. The dollar portfolio weights in rank space $w_{(1),t}, w_{(2),t}$, and in name space $w_{1,t}, w_{2,t}$ are calculated based on capitalization processes in (a) following the intraday rebalancing strategy similar to Fig. 8. The maximum divergence decreases as rebalancing interval increases, aiding risk control. **(c1-c3)** The cumulative latency cost required to compensating divergences at the rebalancing points. Each point of rebalancing incurs a latency cost equal to the divergence between the total dollar portfolio weights in rank space and name space. **(d1-d3)** The cumulative cost from the bid-ask spread due to active trading at each rebalancing point. **(e1-e3)** The cumulative transaction cost. The cumulative transaction costs are the sum of latency costs and transaction costs. The medium rebalancing interval results in the lowest transaction costs while maintaining a manageable divergence, illustrating the importance of choosing an appropriate rebalancing interval.

E Benchmark parametric model

Our parametric model serves as the benchmark for DNNs comparison and follows closely to the framework proposed by Avellaneda and Lee[3] and refined by Yeo and Papanicolaou[28]. This model applies to both name space and rank space, depending on whether x_t^L or $\tilde{x}_t^L$ is chosen as the input. We first fit x_t^L to an OU process X_t governed by the stochastic differential equation

$$dX_t = \frac{1}{\tau}(\mu - X_t)dt + \sigma dB_t, \tag{E.1}$$

where τ is the mean-reverting time, μ is the long-term average of X_t, B_t is the standard Brownian motion, and σ is its volatility. Subsequently, we calculate the trading signal in name space

$$s_{i,t}^{\text{OU}} = \frac{x_{i,t} - \hat{\mu}_i}{\hat{\sigma}_i}, \tag{E.2}$$

where $\hat{\mu}_i, \hat{\sigma}_i$ are the maximum likelihood estimator of μ and σ[3] and $x_{i,t}$ is the terminal cumulative residual return at time t,

$$x_{i,t} = \sum_{j=1}^{L} \epsilon_{i,t-L+j}. \tag{E.3}$$

We also include the estimated mean-reverting time $\hat{\tau}$ to effectively filter the trading opportunities[28]. The details of parameter estimation are presented in the Appendix. We open short/long positions when observing large positive/negative signals and close positions when the trading signals mean-revert close to zero (schematic in Fig. 10). Following the principle, the portfolio weights in residual space, $w_t^{\epsilon|\text{OU,name/rank}}$ become

$$w_{i,t}^{\epsilon|\text{OU,name/rank}} = \begin{cases} -1, & \text{if} \quad w_{i,t-1}^{\epsilon|\text{OU}} = 0, \quad s_{i,t}^{\text{OU}} > c_{\text{thresh-open}}, \quad \hat{\tau}_i < 30 \text{ days} \\ 1, & \text{if} \quad w_{i,t-1}^{\epsilon|\text{OU}} = 0, \quad s_{i,t}^{\text{OU}} < -c_{\text{thresh-open}}, \quad \hat{\tau}_i < 30 \text{ days} \\ 1, & \text{if} \quad w_{i,t-1}^{\epsilon|\text{OU}} = 1, \quad s_{i,t}^{\text{OU}} > c_{\text{thresh-close}}, \quad \hat{\tau}_i < 30 \text{ days} \\ -1, & \text{if} \quad w_{i,t-1}^{\epsilon|\text{OU}} = -1, \quad s_{i,t}^{\text{OU}} > c_{\text{thresh-close}}, \quad \hat{\tau}_i < 30 \text{ days} \\ 0, & \text{otherwise} \end{cases} \tag{E.4}$$

For our back-testing, the parameters are set as follows:

$$c_{\text{thresh-open}} = 1.25, \quad c_{\text{thresh-close}} = 0.5, \tag{E.5}$$

in accordance with [3, 28]. After calculating $w_t^{\epsilon|\text{OU,name/rank}}$, the conversion to portfolio weights in equity space, $w_t^{R|\text{OU,name/rank}}$, straightforwardly follow the (3.4) in name space

$$w_t^{R|\text{OU,name}} = \Phi_t^T w_t^{\epsilon|\text{OU,name}} \tag{E.6}$$

and from (3.8) in rank space,

$$w_t^{R|\text{OU,rank}} = \tilde{\Phi}_t^T w_t^{\epsilon|\text{OU,name}}. \tag{E.7}$$

The practical implementation of the parametric model is summarized in Algorithm 2 along with a schematic in panel (d1, e1) in Fig. 1.

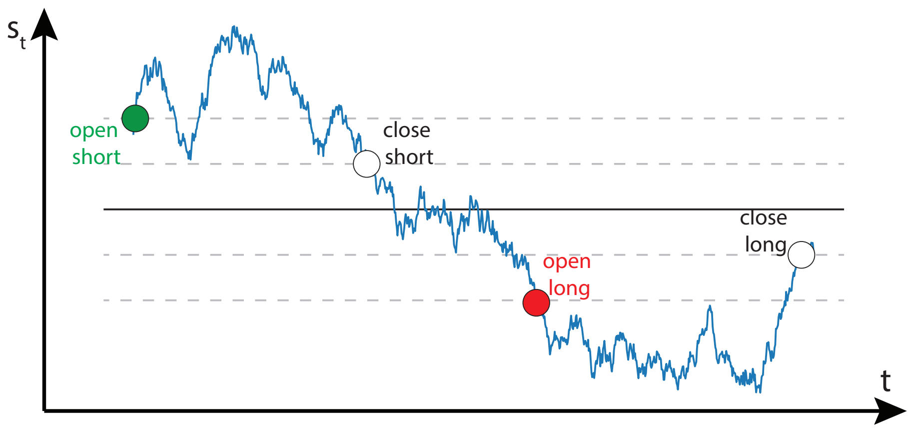

Figure 10: **Schematic for the parametric model.** The blue line shows the trading signal s_t. We open short/long positions when observing large positive/negative signals and close positions when the trading signals mean-revert to zero.

F Non-parametric analysis on mean-reversion of residual returns

The superior mean-reverting behavior in rank space is further demonstrated by comparing the empirical distribution of normalized cumulative residual returns, $\tilde{x}_t^L$ calculated from name space and rank space. We define the normalized cumulative residual returns as follows:

$$\hat{x}_t^L = (\hat{x}_{t-L+1}, \hat{x}_{t-L+2}, ..., \hat{x}_t), \quad \text{where } \hat{x}_{t-L+\alpha} = \frac{1}{\hat{\sigma}_t^L \sqrt{\alpha}} \sum_{j=1}^{\alpha} \epsilon_{t-L+j} \tag{F.1}$$

, where $\hat{\sigma}_t^L$ is the estimated standard deviation of $\{\epsilon_{t-L+j}\}_{j=1}^L$. Suppose the residual returns follow uncorrelated, normal distribution, i.e. $\{\epsilon_{t-L+j}\}_{j=1}^L \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_L)$, the derived cumulative residual returns x_t^L will follow a standard Brownian motion and the normalized cumulative residual return defined in (F.1) will be normally distributed, i.e. $\hat{x}_{t-L+\alpha} \sim \mathcal{N}(0,1), \forall \alpha = 1, 2, ...L$. Consequently, it will serve as a measure of mean-reversion of the difference in probability density function (p.d.f.) between the empirical observations on market and the normal distribution,

$$\text{p.d.f.}(\hat{x}_{t-L+\alpha}) - \frac{1}{\sqrt{2\pi}} \exp(-\frac{\hat{x}_{t-L+\alpha}^2}{2}), \quad \alpha = 1, 2, ..., L \tag{F.2}$$

The difference is accessed over a series of five-year periods from 1991 to 2022 for both name space (Fig. 11(a1-a6)) and rank space (Fig. 11(b1-b6)). A more concentrated distribution of $\hat{x}_{t-L+\alpha}$ than Brownian motion indicates good mean-reverting behavior, especially for large α. This is particularly evident in rank space, where a robust dominance of red color in the heatmaps for large α regime (highlighted in dashed boxes in Fig. 11) underscores the concentrated nature of $\hat{x}_t^L$ in rank space. Such behavior provides critical evidence of a more robust mean-reversion of residual returns in rank space. In stark contrast, the similar dominance by red color in name space was evident in 1990s (Fig. 11(a1, a2)), but progressively deteriorated since 2000s (Fig. 11(a5-a6)), and finally disappears after 2010s (Fig. 11(a5-a6)). This marks the deterioration of the mean-reversion of x_t^L in name space after the 2010s, echoing the failure of profiting from conventional statistical arbitrage strategies after 2010s.

In summary, our non-parametric analysis here demonstrates that the rank space exhibits more robust mean-reverting behavior compared to name space, echoing the parametric analysis on mean-reverting time in the main text.

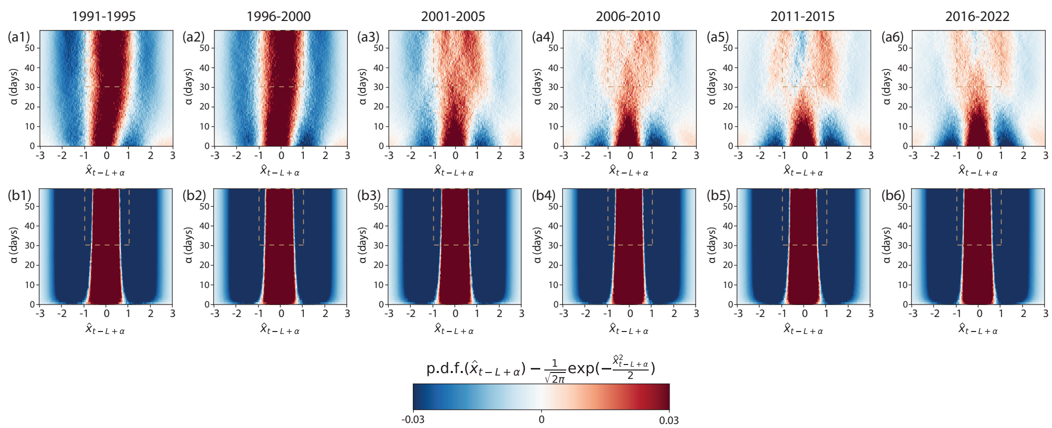

Figure 11: **Empirical distributions of normalized cumulative residual returns: name space versus rank space.** The cumulative residual returns x_t^L are normalized according to (F.1). This normalization facilitates the conversion of comparisons between trajectories of x_t^L and Brownian motion into comparisons of the probability density of the empirical distribution of $\hat{x}_t^L$ against the normal distribution. Consequently, we present the difference between the empirical probability density of $\hat{x}_t^L$ and standard normal distribution. The empirical probability density of $\hat{x}_t^L$ is evaluated across a series of five-year periods from 1991 (left) to 2022 (right) in both **(a1-a6)** name space and **(b1-b6)** rank space. The dashed brown box highlights the critical regime where a dominating red color indicates a more concentrated distribution of $x_{t-L+\alpha}$ for large α. This concentration signals superior mean-reversion capabilities, particularly evident in rank space throughout the last thirty years. Furthermore, the distribution of $\hat{x}_t^L$ in name space evolves significantly, indicating a progressively deteriorating mean-reversion in name space after 2010. This echoes the relatively poor performance in our backtesting. The stark contrast in $\hat{x}_t^L$ supports the strategic advantage of operating in rank space for statistical arbitrage.

G Portfolio performance

G.1 Annualized return, volatility, and Sharpe ratio

We calculate the annualized return, volatility, and Sharpe ratio derived from the PnL V_t in Fig. 5. The annualized summary statistics without and with transaction costs are presented in Table 1 and Table 2, respectively, where we consider corresponding portfolio weights w_t^R calculated by four scenarios: (i) the parametric benchmark model in name space in panel (a), (ii) the parametric benchmark model in rank space in panels (b,c), (iii) DNNs in name space in panel (d), and (iv) DNNs in rank space in panels (e, f).

year	Name space parametric model			rank space parametric model			name space neural networks			rank space neural networks		
	return	vol	SR	return	vol	SR	return	vol	SR	return	vol	SR
2007	2.87%	0.02	1.38	8.52%	0.04	1.90	-8.37%	0.10	-0.87	79.41%	0.14	5.62
2008	7.42%	0.04	1.70	25.44%	0.06	3.96	-6.98%	0.13	-0.52	239.88%	0.38	6.36
2009	7.01%	0.03	2.02	27.81%	0.08	3.57	3.91%	0.13	0.29	414.09%	0.35	11.72
2010	-0.49%	0.02	-0.24	31.73%	0.05	6.21	-0.02%	0.08	0.00	222.37%	0.19	11.41
2011	1.91%	0.02	0.85	40.14%	0.06	7.12	12.84%	0.08	1.67	126.84%	0.22	5.76
2012	-0.40%	0.02	-0.22	41.06%	0.05	8.20	7.35%	0.07	1.00	162.37%	0.20	8.20
2013	1.19%	0.02	0.70	27.92%	0.05	5.74	8.34%	0.07	1.14	289.73%	0.21	13.96
2014	3.65%	0.02	2.07	43.82%	0.05	8.84	-3.24%	0.07	-0.49	168.89%	0.14	12.07
2015	0.81%	0.02	0.41	41.78%	0.06	7.44	0.71%	0.08	0.09	137.95%	0.19	7.10
2016	2.79%	0.02	1.43	61.86%	0.07	9.51	8.58%	0.10	0.90	293.85%	0.27	11.02
2017	1.56%	0.02	0.93	30.58%	0.04	7.04	9.88%	0.07	1.35	208.30%	0.17	12.10
2018	3.07%	0.02	1.44	27.78%	0.05	5.71	3.67%	0.06	0.60	151.91%	0.19	7.83
2019	4.50%	0.02	2.44	41.42%	0.05	8.39	-6.84%	0.06	-1.07	175.91%	0.20	8.59
2020	1.56%	0.04	0.39	25.06%	0.09	2.89	1.24%	0.09	0.14	307.81%	0.28	10.84
2021	-0.67%	0.02	-0.28	37.60%	0.06	6.21	-2.32%	0.08	-0.30	177.60%	0.25	7.11
2022	1.05%	0.03	0.36	36.79%	0.07	5.53	28.84%	0.09	3.29	146.86%	0.29	5.00
Avg	2.36%	0.02	0.96	34.33%	0.06	6.14	3.60%	0.08	0.45	206.49%	0.23	9.04

Table 1: **Portfolio performance without transaction costs.** The portfolios in rank space consistently outperform their counterparts in name space, both with the parametric model and neural networks. The neural networks improve the portfolio performance in rank space dramatically, in stark contrast with negligible improvements in name space. The contrast echoes with the fact that the neural networks are much more effective in rank space compared to that in name space Fig. 7.

year	Name space parametric model			rank space parametric model			name space neural networks			rank space neural networks		
	return	vol	SR	return	vol	SR	return	vol	SR	return	vol	SR
2007	1.63%	0.02	0.79	-32.51%	0.04	-7.67	-15.51%	0.10	-1.61	26.07%	0.10	2.55
2008	6.02%	0.04	1.38	-43.78%	0.05	-9.58	-14.19%	0.13	-1.06	36.40%	0.18	2.04
2009	5.71%	0.03	1.65	-32.78%	0.04	-8.07	-3.13%	0.13	-0.23	48.97%	0.13	3.67
2010	-1.69%	0.02	-0.82	-32.83%	0.03	-12.68	-7.25%	0.08	-0.91	43.14%	0.10	4.32
2011	0.71%	0.02	0.31	-32.45%	0.03	-12.02	5.10%	0.08	0.66	14.32%	0.10	1.45
2012	-1.52%	0.02	-0.84	-26.53%	0.02	-11.84	-0.20%	0.07	-0.03	20.41%	0.08	2.42
2013	0.02%	0.02	0.01	-25.14%	0.02	-11.16	0.21%	0.07	0.03	52.51%	0.10	5.37
2014	2.45%	0.02	1.39	-22.52%	0.02	-9.74	-10.25%	0.07	-1.55	35.55%	0.07	4.76
2015	-0.31%	0.02	-0.16	-20.12%	0.03	-7.38	-6.89%	0.08	-0.89	22.82%	0.10	2.32
2016	1.65%	0.02	0.85	-17.68%	0.03	-5.45	0.43%	0.10	0.05	56.09%	0.13	4.31
2017	0.36%	0.02	0.22	-21.38%	0.02	-9.28	2.62%	0.07	0.36	49.00%	0.09	5.16
2018	1.86%	0.02	0.87	-28.83%	0.03	-11.45	-4.16%	0.06	-0.68	27.94%	0.10	2.81
2019	3.34%	0.02	1.81	-21.49%	0.02	-8.92	-13.52%	0.06	-2.12	34.13%	0.10	3.38
2020	0.21%	0.04	0.05	-42.12%	0.05	-9.02	-5.62%	0.09	-0.62	56.62%	0.14	4.14
2021	-2.06%	0.02	-0.86	-36.82%	0.03	-13.27	-9.69%	0.08	-1.27	31.14%	0.13	2.47
2022	-0.16%	0.03	-0.06	-33.01%	0.03	-11.71	19.19%	0.09	2.19	15.69%	0.12	1.26
Avg	1.14%	0.02	0.41	-29.37%	0.03	-9.95	-3.93%	0.08	-0.48	35.68%	0.11	3.28

Table 2: **Portfolio performance with transaction costs.** The portfolio performances in rank space degrade dramatically, for both the parametric model and the neural networks. The substantial degradation arises from the substantial costs associated with realizing rank return in continuous time limit through intraday rebalancing. Nevertheless, the portfolio calculated by neural networks in rank space still yields good results, as the significant transaction costs are compensated by the impressive returns and Sharpe ratio in Table 1, column 4. We choose 2 basis points to account for the transaction costs from bid-ask spread.

G.2 Dollar neutrality

We characterize the long or short proportion of the portfolio weights in equity space, $\sum_{i:w_{i,t}^R>0} w_{i,t}$ or $\sum_{i:w_{i,t}^R<0} w_{i,t}$ and the dollar neutrality, $\frac{\sum_i w_{i,t}^R}{\sum_i |w_{i,t}^R|}$. The results are presented in Fig. 12(a1-d1) and Fig. 12(a2-d2), where we consider w_t^R calculated by four scenarios: (i) the parametric benchmark model in name space (Fig. 12(a1, a2), (ii) DNNs in name space (Fig. 12(b1, b2), (iii) the parametric benchmark model in rank space (Fig. 12(c1, c2), (iv) DNNs in rank spce (Fig. 12(d1, d2)). Notably, the long or short proportion of w_t^R by neural networks (Fig. 12(b1, d1)) is much more volatile than those by the parametric model (Fig. 12(a1, c1)), as a result of flexible leverage adopted by neural networks. However, the dollar neutrality is satisfied on average thanks to the market neutrality of the portfolios.

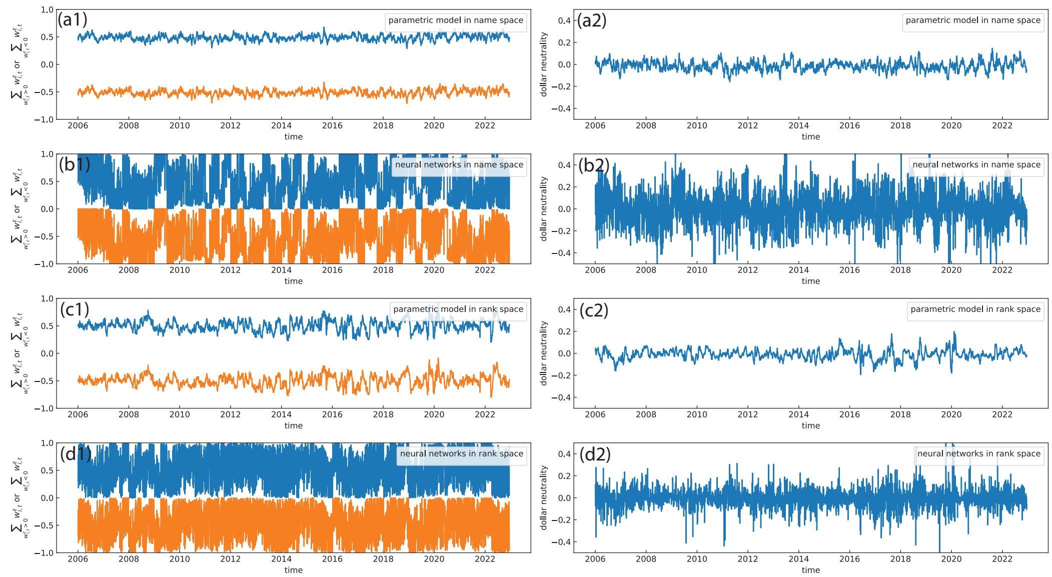

Figure 12: **Portfolio weights in residual return and dollar neutrality. (a1-d1)** The temporal dependence of average long/short portfolio weights in residual space, $\sum_{w_{i,t}^\epsilon>0} w_{i,t}^\epsilon$ and $\sum_{w_{i,t}^\epsilon<0} w_{i,t}^\epsilon$. **(a2-d2)** The deviation from dollar neutrality measured by $\frac{\sum_i w_{i,t}^R}{\sum_i |w_{i,t}^R|}$. We consider four scenarios: (a1-a2) parametric model in name space; (b1-b2) neural networks in name space; (c1-c2) parametric model in rank space; (d1-d2) neural networks in rank space.

G.3 Dependence on transaction costs

We present the sensitivity of portfolio performance to transaction costs. We show the PnL with different transaction cost factor η in Fig. 13(a) and the corresponding Sharpe ratio in Fig. 13. The current strategy shows significant sensitivity to the transaction cost factor and stops to profit with $\eta = 5$ basis points. A more effective strategy to realize return in rank space $\tilde{r}_t$ will help the strategy more immune to transaction costs.

G.4 A characteristic time between rank switching

The rebalancing interval, $\mathcal{T}$, turns out to be a crucial parameter for statistical arbitrage portfolios in rank space. To demonstrate, we present the calculated PnL with varying rebalancing intervals in Fig. 14(a), where the portfolio weights are calculated from the neural networks in rank space. The associated average Sharpe ratio and terminal PnL as a function of the rebalancing interval are summarized in Fig. 14(b), with both peaking at 225 minutes.

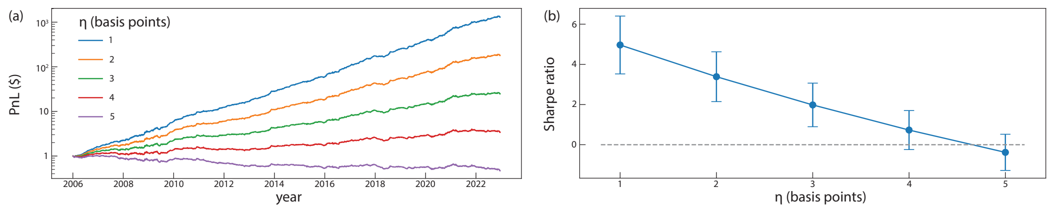

Figure 13: **PnL dependence on transaction cost factor η. (a)** The PnL with varing levels of the transaction cost factor η. The underlying portfolio weights are derived from the neural networks in rank space. **(b)** The average Sharpe ratio from 2006 to 2022 at different values of η derived from (a). The strategy stops to profit with 5 basis points transaction costs due to substantial costs associated with realizing rank returns in continuous time limit. The significant change in Sharpe ratio under varying transaction costs underscores the strategy's sensitivity to transaction costs and motivates ongoing improvements in "trading ranks".

Here, we delve into the rationale behind the optimal 225-minute interval, starting with an examination of two proximate capitalization processes modeled as Brownian motions, $c_{1,t}$ and $c_{2,t}$ (Fig. 14(c)). Given that transaction costs from intraday rebalancing primarily arise from rank swaps in capitalization (section 2.3), we measure the cumulative time $\Lambda(t)$ that the capitalization processes cross (Fig. 14(c), red line),

$$\Lambda(t) = \lim_{\delta \downarrow 0} \int_0^t \mathbf{1}_{\{|c_{1,\tau} - c_{2,\tau}| \leq \delta\}} d\tau \tag{G.1}$$

The rank-swapping interval λ is defined as the time between the cross of capitalization processes, or equivalently, the increments of $\Lambda(t)$. This classical interacting Brownian system features two characteristic regimes: (i) the idle regime, where the two capitalization processes are distant, maintaining constant $\Lambda(t)$ with prolonged λ; (ii) the collision regime (highlighted in brown shaded area in Fig. 14(c)), where the two capitalization processes stay close, leading to rapid increases in $\Lambda(t)$ and short λ. We show the empirical distribution of λ on real market in Fig. 14(d), where the small λ values arise from the collision regime and larger λ values from the idle regime, following approximately an exponential distribution as a typical signature for standard Brownian particle systems. The 225 minutes is situated at the intersection of the two regimes, establishing it as a characteristic time for rank switching.

In the detailed analysis of the intraday rebalancing in (3.14) and Fig. 9, the transaction costs arise from (i) latency costs due to delayed reactions post-rank-swapping, and (ii) costs from bid-ask spreads incurred during active trading. For the collision regime in Fig. 14(c), it is preferable to delay trading to minimize bid-ask spread costs. Conversely, in the idle regime, immediate trading is preferable to reduce latency costs. The 225-minute interval effectively differentiates these regimes, thus optimizing overall transaction costs.

The discussion above highlights the challenge in trading ranks – discerning between the collision and idle regimes and trading at their intersection. Our current intraday rebalancing approach crudely harnesses average behavior of U.S. equity market, and leaves considerable scope for enhancement that we will follow up on in subsequent research papers.

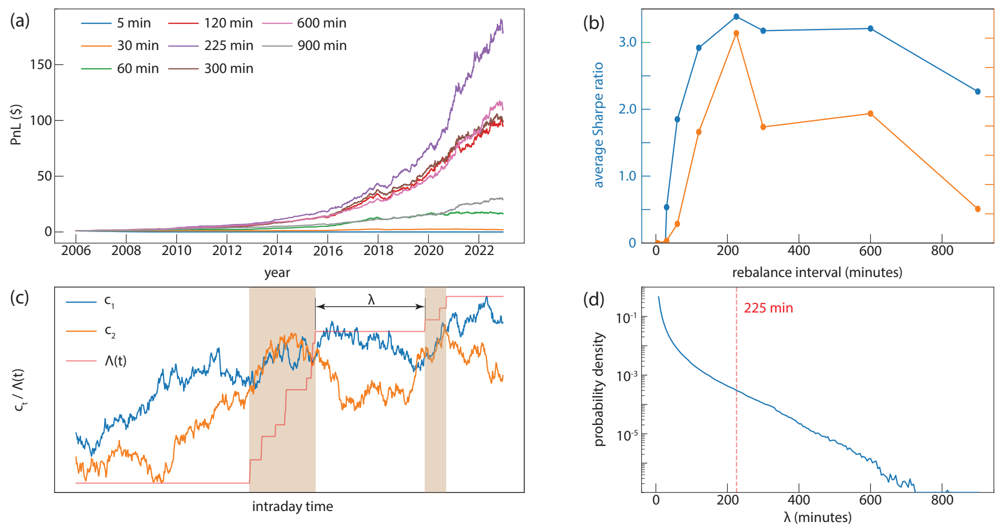

Figure 14: **PnL dependence on rebalancing interval and characteristic rank-swapping time.** **(a)** The PnL across various intraday rebalancing intervals $\mathcal{T}$ to realize the rank return in continuous time limit $\tilde{r}_t$. The underlying portfolio weights are derived from the neural networks in rank space. **(b)** Averaged Sharpe ratio between 2006 and 2022 and the terminal PnL as functions of rebalance intervals from (a). Both metrics peak at $\mathcal{T} = 225$ minutes. **(c)** Schematic representation of two capitalization processes,c_1 and c_2. The $\Lambda(t)$ measures the cumulative time that the capitalization processes c_1 and c_2 cross. Rank switching time λ is the interval between the increments in local time. This stochastic system has two characteristic regimes: (i) the idle regime, where the two capitalization processes are distant, maintaining constant $\Lambda(t)$ with prolonged λ; (ii) the collision regime (highlighted in brown shaded area in Fig. 14(c)), where the two capitalization processes stay close, leading to rapid increases in $\Lambda(t)$ and short λ. **(e)** The empirical distributions of rank-swapping time τ based on the intraday market data. Low (high) λ arises from the "idle" ("collision") regime. The red dashed line marks the optimal rebalancing interval, $\mathcal{T} = 225$ minutes, positioned at the intersection between the idle and collision regimes, suggesting it is a balanced choice for minimizing transaction costs while responding effectively with rank-swapping events.

References

[1] Fernando Acero, Parisa Zehtabi, Nicolas Marchesotti, Michael Cashmore, Daniele Magazzeni, and Manuela Veloso. Deep reinforcement learning and mean-variance strategies for responsible portfolio optimization. *arXiv:2403.16667*, 2024.

[2] Dogu Araci. Finbert: Financial sentiment analysis with pre-trained language models. arxiv 2019. *arXiv preprint arXiv:1908.10063*, 2019.

[3] Marco Avellaneda and Jeong-Hyun Lee. Statistical arbitrage in the US equities market. *Quantitative Finance*, 10(7):761–782, 2010.

[4] Marco Avellaneda, Brian Healy, Andrew Papanicolaou, and George Papanicolaou. Principal eigenportfolios for US equities. *SIAM Journal on Financial Mathematics*, 13(3):702–744, 2022.

[5] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv:1607.06450*, 2016.

[6] Adrian D Banner and Raouf Ghomrasni. Local times of ranked continuous semimartingales. *Stochastic Processes and their Applications*, 118(7):1244–1253, 2008.

[7] Adrian D Banner, Robert Fernholz, and Ioannis Karatzas. Atlas models of equity markets. *Annals of applied probability*, 15(4):2296–2330, 2005.

[8] Howard C Berg. *Random walks in biology*. Princeton University Press, 1993.

[9] Jean-Philippe Bouchaud and Marc Potters. Financial applications of random matrix theory: a short review. *arXiv preprint arXiv:0910.1205*, 2009.

[10] Álvaro Cartea, Sebastian Jaimungal, and José Penalva. *Algorithmic and high-frequency trading*. Cambridge University Press, 2015.

[11] E Robert Fernholz and E Robert Fernholz. *Stochastic portfolio theory*. Springer, 2002.

[12] Cesar Garcia-Diaz. Sociophysics: A physicist's modeling of psycho-political phenomena (understanding complex systems), 2013.

[13] Shihao Gu, Bryan Kelly, and Dacheng Xiu. Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5):2223–2273, 2020.

[14] Jorge Guijarro-Ordonez, Markus Pelger, and Greg Zanotti. Deep learning statistical arbitrage. *arXiv:2106.04028*, 2021.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[16] Brian Healy, Y.-F. Li, A Papanicolaou, and P Papanicolaou. The data stabilizing transformation from named to ranked equity returns. *working paper*, 2025.

[17] Blanka Horvath, Aitor Muguruza, and Mehdi Tomas. Deep learning volatility: a deep neural network perspective on pricing and calibration in (rough) volatility models. *Quantitative Finance*, 21(1):11–27, 2021.

[18] Tomoyuki Ichiba, Vassilios Papathanakos, Adrian Banner, Ioannis Karatzas, and Robert Fernholz. Hybrid atlas models. *Annals of applied probability*, 21(2):609–644, 2011.

[19] Taewook Kim and Ha Young Kim. Optimizing the pairs-trading strategy using deep reinforcement learning with trading and stop-loss boundaries. *Complexity*, 2019(1):3582516, 2019.

[20] Tim Siu-tang Leung and Xin Li. *Optimal mean reversion trading: Mathematical analysis and practical applications*, volume 1. World Scientific, 2015.

[21] Gerald D Mahan. *Many-particle physics*. Springer Science & Business Media, 2013.

[22] John M Mulvey, Yifan Sun, Mengdi Wang, and Jing Ye. Optimizing a portfolio of mean-reverting assets with transaction costs via a feedforward neural network. *Quantitative Finance*, 20(8):1239–1261, 2020.

[23] Szilárd Pafka and Imre Kondor. Noisy covariance matrices and portfolio optimization ii. *Physica A: Statistical Mechanics and its Applications*, 319:487–494, 2003.

[24] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[25] Agnès Tourin and Raphael Yan. Dynamic pairs trading using the stochastic control approach. *Journal of Economic Dynamics and Control*, 37(10):1972–1981, 2013.

[26] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv:1607.08022*, 2016.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[28] Joongyeub Yeo and George Papanicolaou. Risk control of mean-reversion time in statistical arbitrage. *Risk and Decision Analysis*, 6(4):263–290, 2017.

[29] Jiongmin Yong and Xun Yu Zhou. *Stochastic controls: Hamiltonian systems and HJB equations*, volume 43. Springer Science & Business Media, 2012.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in Appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope. We claim that (1) operating statistical arbitrage in rank space significantly improves performance over name space when using deep neural networks (DNNs), and (2) this improvement is attributed to the more structured market dynamics and enhanced mean-reversion properties in rank space. These claims are systematically validated through empirical backtesting on U.S. equities from 2007 to 2022, detailed ablation studies, and analysis of the learned neural network behaviors. The paper remains focused on demonstrating these two core contributions without over-claiming broader applicability beyond the presented setup.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.

- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper explicitly discusses the limitations of the work in the *limitations and discussion* section. We highlight that the current strategy is sensitive to transaction costs due to frequent intraday rebalancing required to realize rank returns, which may limit its immediate practicality in less liquid markets. We also provide the framework for future optimization of execution strategies to address this limitation. Additionally, while we demonstrate our method in financial markets, we recognize that broader generalization to other many-particle systems is suggested but not empirically validated in this work.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: For each theoretical result, we provide the full set of assumptions and complete proofs in the Appendix. In particular, we include (i) a detailed proof of the market neutrality property of the constructed portfolios under the given market decomposition framework, (ii) a hybrid-Atlas model that motivates operating statistical arbitrage in rank space. All mathematical arguments are presented carefully, and the proofs are fully self-contained.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in Appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide full disclosure of the information needed to reproduce the main experimental results. Detailed pseudocode for all key algorithms is included in the Appendix. In addition, we have uploaded all Python code necessary for calculating the results to a public GitHub repository. However, the financial market data used in this study cannot be shared publicly: the daily security data are sourced from CRSP, and the high-frequency intraday data are sourced from Polygon.io, both of which are under licensing agreements that prohibit redistribution. Despite this, we provide sufficient algorithmic and implementation details for independent reproduction using equivalent market datasets.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide open access to the full Python code necessary to reproduce the main experimental results on GitHub repository, along with detailed instructions and pseudocode in the supplemental material. However, we cannot provide open access to the data due to licensing restrictions: the daily security data are sourced from CRSP and the high-frequency intraday data from Polygon.io, both of which are proprietary and not publicly distributable. Despite this, all necessary procedures and implementation details are disclosed to enable faithful reproduction using equivalent datasets.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide all necessary training and testing details. The pseudocode in the Appendix outlines the complete experimental workflow, including data preparation, model training, and evaluation. We also specify the full architecture of the neural network in the appendix, including convolutional and transformer layers, and describe the training procedure, including input features, objective function (mean-variance optimization), and training window sizes. Hyperparameters such as the number of channels, kernel sizes, number of attention heads, and dropout rates are explicitly stated. Our provided pseudocode and implementation details allow faithful reproduction.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in Appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [NA]

Justification: We report performance using average annual returns and Sharpe ratios across multiple years (2007–2022), with detailed year-by-year statistics provided in the Appendix.

Since our experiments are based on real historical market data rather than simulations, traditional error bars or confidence intervals over repeated trials are not applicable. Instead, we present the variability of results across different years to convey the robustness and statistical significance of our findings.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide sufficient information on the computational resources used in the Appendix. We specify the hardware configuration, including the type of GPU and CPU, memory capacity, and details of the training process. This information enables readers to estimate the computational requirements needed to reproduce the experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research presented in the paper fully conforms to the NeurIPS Code of Ethics. Our work involves only publicly available financial market data (subject to standard licensing restrictions) and does not involve human subjects, private information, or sensitive demographic attributes. We adhere to principles of transparency, reproducibility, and responsible reporting throughout the paper.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Our work introduces a rank-based representation of market dynamics that improves the learning efficiency of deep neural networks in noisy environments, with potential applications beyond finance to complex systems in physics, biology, and social science. By providing a more structured and stable modeling framework, this approach may facilitate new insights into the behavior of many-particle systems. However, in financial contexts, high-frequency statistical arbitrage strategies could contribute to liquidity fragmentation or increase market volatility if deployed without appropriate safeguards. Our method is also sensitive to transaction costs, which may limit its accessibility to participants with unequal technological resources. We encourage future research to explore both beneficial applications and responsible deployment of rank-based learning methods.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our work does not involve the release of pretrained models, scraped datasets, or generative models that carry a high risk of misuse. The models we develop are applied to historical financial market data under standard licensing restrictions, and no sensitive or private information is used. As such, no additional safeguards beyond responsible reporting and licensing compliance are necessary.

    Guidelines:

    - The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All external assets used in the paper are properly credited and their licenses respected. We clearly state that the daily financial data are sourced from CRSP and the high-frequency intraday data are obtained from Polygon.io, both under institutional licensing agreements. We also acknowledge that these datasets cannot be publicly redistributed due to licensing restrictions. Any public libraries or frameworks used for model development and training (such as standard deep learning libraries) are cited where appropriate.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The primary new asset introduced in the paper is the implementation code for our statistical arbitrage framework based on rank-space modeling. We provide detailed pseudocode in the Appendix and release the full Python code with accompanying documentation and instructions on a public GitHub repository. This ensures that others can faithfully reproduce the main results and understand the implementation details.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our research does not involve any crowdsourcing experiments, human subjects, or participant studies. All experiments are conducted on historical financial market data without human interaction.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our research does not involve any study participants or human subjects. Therefore, no risks to participants exist, no disclosures were necessary, and no Institutional Review Board (IRB) approvals were required.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.